*A GROUP PROJECT REPORT ON*

*“SECURITY ANALYSIS OF TEA ALGORITHM*

*USING LIGHT WEIGHT CRYPTOGRAPHY”*

**S**ubmitted to the Dept of Information Technology, SNIST

In the Partial Fulfillment of the Academic Requirements for the Award of

B.Tech(IT) Under JNTU .

By

PAKALA POOJITHA(15311A12J7)

GIVALLA NIKHIL KUMAR(15311A12K3)

Md.ASIF BABA(15311A12K8)

Under the Guidance of

Mr. Nalamala Chaitanya Kumar

Associate Professor

SNIST



**Department of Information Technology**

*School of Computer Science & Information*

**Sreenidhi Instutition of Science & Technology**

*Yamnampet Ghatkesar Mandal RR-Dist Hyderabad 501301*

*Affiliated to*

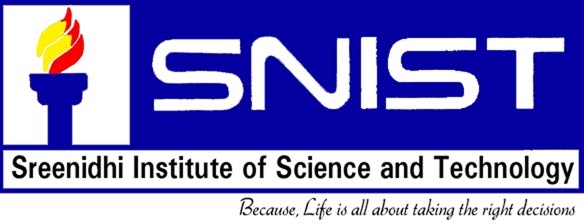
**Jawaharlal Nehru Technological University**

Hyderabad – 500 085 2017

**Department of Information Technology**

**School of Computer Science and Informatics**

**Sreenidhi Institute of Science and Technology**

****

**CERTIFICATE**

This is certify that the Technical Seminar report on ”SECURITY ANALYSIS OF TEA ALGORITHM USING LIGHT WEIGHT CRYPTOGRAPHY” is a bonafide work carried out by “PAKALA POOJITHA(15311A12J7)GIVALLA NIKHIL KUMAR(15311A12K3) Md.ASIF BABA(15311A12K8)” in the partial fulfillment for the award of B.Tech degree in Information Technology, Sreenidhi Institute of Science and Technology, Hyderabad affiliated to Jawaharlal Nehru Technology University Hyderabad, under our guidance and supervision.

This results embodied in the project work have not been submitted to any other university or institute for the award of any degree or diploma.

HEAD OF DEPARTMENT

PROF.V.V.S.S.S.BALARAM

Professor &head

Department of

Information Technology

SNIST

INTERNAL GUIDE Mr.N.Chaitanya Kumar

Associate Professor

**ACKNOWLEDGEMENTS**

we would like to express my sincere thanks to **Dr.P.Narasimha Reddy**, Director **Dr.K.Sumanth**, Principal, **Dr.UmaKanth Sastry**, Director School of Computer Science and Informatics**, Prof.V.V.S.S.S.Balaram**, Head of Department of Information Technology, Project Guide **N.Chaitanya Kumar**, **SreeNidhi Institute** **of Science and Technology**, Hyderabad for permitting me to do my project work at Sreendihi Institute of Science and Technology -Hyderabad.

Finally, we would also like to thank the people who have directly or indirectly helped me and my parents, friends and relatives for their cooperation in completing the project work.

We would like to express our immense gratitude and sincere thanks to our Internal guide **Mr.N.Chaitanya kumar**,Associate professor for your guidance and suggestions and encouragement in completing our group project within the stipulated time.

ABSTRACT

SECURITY ANALYSIS OF TEA ALGORITHM USING

LIGHT WEIGHT CRYPTOGRAPHY

The Tiny Encryption Algorithm (TEA) is a cryptographic algorithm designed to minimize memory footprint and maximize speed. It is a Feistel type cipher that uses operations from mixed (orthogonal) algebraic groups. This research presents the cryptanalysis of the Tiny Encryption Algorithm. In this research we inspected the most common methods in the cryptanalysis of a block cipher algorithm. TEA seems to be highly resistant to differential cryptanalysis, and achieves complete diffusion (where a one bit difference in the plaintext will cause approximately 32 bit differences in the cipher text) after only six rounds. Time performance on a modern desktop computer or workstation is very impressive This paper aims at inspecting few of the methods in the cryptanalysis of a TEA algorithm and to observe the possibilities of breaking the cipher text by attacker.Analysis of how many encryptions and how many known plain text and cipher text are required for attacker to succeed Such that we can ensure to which extend it is secured and where we can use this and where we cant.

INTERNAL GUIDE BY

Mr VIJAY BHASKER Pakala Poojitha(15311A12J7)

(Associate Professor) Givalla Nikhil(15311A12K3)

Md.Asif Baba(15311A12K8)

**INTRODUCTION**

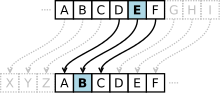
**Cryptography** or **cryptology** (from [Greek](https://en.wikipedia.org/wiki/Ancient_Greek) [κρυπτός](https://en.wiktionary.org/wiki/en:%CE%BA%CF%81%CF%85%CF%80%CF%84%CF%8C%CF%82" \o "wikt:en:κρυπτός) *kryptós*, "hidden, secret"; and[γράφειν](https://en.wiktionary.org/wiki/en:%CE%B3%CF%81%CE%AC%CF%86%CF%89#Ancient_Greek) *graphein*, "writing", or [-λογία](https://en.wiktionary.org/wiki/en:-%CE%BB%CE%BF%CE%B3%CE%AF%CE%B1#Greek) [*-logia*](https://en.wikipedia.org/wiki/-logy), "study", respectively) is the practice and study of techniques for  [secure communication](https://en.wikipedia.org/wiki/Secure_communication) in the presence of third parties called [adversaries](https://en.wikipedia.org/wiki/Adversary_(cryptography)). More generally, cryptography is about constructing and analyzing [protocols](https://en.wikipedia.org/wiki/Communications_protocol) that prevent third parties or the public from reading private messages; various aspects in [information security](https://en.wikipedia.org/wiki/Information_security) such as data [confidentiality](https://en.wikipedia.org/wiki/Confidentiality" \o "Confidentiality),[data integrity](https://en.wikipedia.org/wiki/Data_integrity), [authentication](https://en.wikipedia.org/wiki/Authentication), and [non-repudiation](https://en.wikipedia.org/wiki/Non-repudiation) are central to modern cryptography. Modern cryptography exists at the intersection of the disciplines of[mathematics](https://en.wikipedia.org/wiki/Mathematics), [computer science](https://en.wikipedia.org/wiki/Computer_science), and [electrical engineering](https://en.wikipedia.org/wiki/Electrical_engineering). Applications of cryptography include [military communications](https://en.wikipedia.org/wiki/Military_communications), [electronic commerce](https://en.wikipedia.org/wiki/Electronic_commerce), [ATM cards](https://en.wikipedia.org/wiki/Automated_teller_machine), and[computer passwords](https://en.wikipedia.org/wiki/Password).

Cryptography prior to the modern age was effectively synonymous with [*encryption*](https://en.wikipedia.org/wiki/Encryption), the conversion of information from a readable state to apparent [nonsense](https://en.wikipedia.org/wiki/Nonsense). The originator of an encrypted message (Alice) shared the decoding technique needed to recover the original information only with intended recipients (Bob), thereby precluding unwanted persons (Eve) from doing the same. The cryptography literature often uses Alice ("A") for the sender, Bob ("B") for the intended recipient, and Eve ("[eavesdropper](https://en.wikipedia.org/wiki/Eavesdropper)") for the adversary. Since the development of [rotor cipher machines](https://en.wikipedia.org/wiki/Rotor_machine) in [World War I](https://en.wikipedia.org/wiki/World_War_I) and the advent of [computers](https://en.wikipedia.org/wiki/Computer) in [World War II](https://en.wikipedia.org/wiki/World_War_II), the methods used to carry out cryptology have become increasingly complex and its application more widespread.

Modern cryptography is heavily based on mathematical theory and computer science practice; cryptographic algorithms are designed around [computational hardness assumptions](https://en.wikipedia.org/wiki/Computational_hardness_assumption), making such algorithms hard to break in practice by any adversary. It is theoretically possible to break such a system, but it is infeasible to do so by any known practical means. These schemes are therefore termed computationally secure; theoretical advances, e.g., improvements in [integer factorization](https://en.wikipedia.org/wiki/Integer_factorization) algorithms, and faster computing technology require these solutions to be continually adapted. There exist [information-theoretically secure](https://en.wikipedia.org/wiki/Information_theoretic_security) schemes that provably cannot be broken even with unlimited computing power—an example is the [one-time pad](https://en.wikipedia.org/wiki/One-time_pad)—but these schemes are more difficult to implement than the best theoretically breakable but computationally secure mechanisms.

The growth of cryptographic technology has raised a number of legal issues in the information age. Cryptography's potential for use as a tool for [espionage](https://en.wikipedia.org/wiki/Espionage) and [sedition](https://en.wikipedia.org/wiki/Sedition) has led many governments to classify it as a weapon and to limit or even prohibit its use and export. In some jurisdictions where the use of cryptography is legal, laws permit investigators to [compel the disclosure](https://en.wikipedia.org/wiki/Key_disclosure_law) of encryption keys for documents relevant to an investigation. Cryptography also plays a major role in[digital rights management](https://en.wikipedia.org/wiki/Digital_rights_management) and [copyright infringement](https://en.wikipedia.org/wiki/Copyright_infringement) of digital media.

## TERMINOLOGY

[](https://en.wikipedia.org/wiki/File:Caesar_cipher_left_shift_of_3.svg)

Alphabet shift ciphers are believed to have been used by [Julius Caesar](https://en.wikipedia.org/wiki/Julius_Caesar)over 2,000 years ago. This is an example with k=3. In other words, the letters in the alphabet are shifted three in one direction to encrypt and three in the other direction to decrypt.

Until modern times, cryptography referred almost exclusively to encryption which is the process of converting ordinary information (called [plaintext](https://en.wikipedia.org/wiki/Plaintext)) into unintelligible text (called [ciphertext](https://en.wikipedia.org/wiki/Ciphertext" \o "Ciphertext)). Decryption is the reverse, in other words, moving from the unintelligible ciphertext back to plaintext. A cipher (or cypher) is a pair of [algorithms](https://en.wikipedia.org/wiki/Algorithm" \o "Algorithm)that create the encryption and the reversing decryption. The detailed operation of a cipher is controlled both by the algorithm and in each instance by a "[key](https://en.wikipedia.org/wiki/Key_(cryptography))". The key is a secret (ideally known only to the communicants), usually a short string of characters, which is needed to decrypt the ciphertext. Formally, a "[cryptosystem](https://en.wikipedia.org/wiki/Cryptosystem)" is the ordered list of elements of finite possible plaintexts, finite possible cyphertexts, finite possible keys, and the encryption and decryption algorithms which correspond to each key. Keys are important both formally and in actual practice, as ciphers without variable keys can be trivially broken with only the knowledge of the cipher used and are therefore useless (or even counter-productive) for most purposes. Historically, ciphers were often used directly for encryption or decryption without additional procedures such as [authentication](https://en.wikipedia.org/wiki/Authentication) or integrity checks. There are two kinds of cryptosystems: [symmetric](https://en.wikipedia.org/wiki/Symmetric-key_algorithm) and [asymmetric](https://en.wikipedia.org/wiki/Public-key_cryptography). In symmetric systems the same key (the secret key) is used to encrypt and decrypt a message. Data manipulation in symmetric systems is faster than asymmetric systems as they generally use shorter key lengths. Asymmetric systems use a public key to encrypt a message and a private key to decrypt it. Use of asymmetric systems enhances the security of communication. Examples of asymmetric systems include RSA ([Rivest-Shamir-Adleman](https://en.wikipedia.org/wiki/Rivest-Shamir-Adleman" \o "Rivest-Shamir-Adleman)), and ECC ([Elliptic Curve Cryptography](https://en.wikipedia.org/wiki/Elliptic_Curve_Cryptography)). Symmetric models include the commonly used AES ([Advanced Encryption Standard](https://en.wikipedia.org/wiki/Advanced_Encryption_Standard)) which replaced the older DES ([Data Encryption Standard](https://en.wikipedia.org/wiki/Data_Encryption_Standard)).

In [colloquial](https://en.wikipedia.org/wiki/Colloquial) use, the term "[code](https://en.wikipedia.org/wiki/Code_(cryptography))" is often used to mean any method of encryption or concealment of meaning. However, in cryptography,code has a more specific meaning. It means the replacement of a unit of plaintext (i.e., a meaningful word or phrase) with a [code word](https://en.wikipedia.org/wiki/Code_word) (for example, "wallaby" replaces "attack at dawn").

[Cryptanalysis](https://en.wikipedia.org/wiki/Cryptanalysis) is the term used for the study of methods for obtaining the meaning of encrypted information without access to the key normally required to do so; i.e., it is the study of how to crack encryption algorithms or their implementations.

Some use the terms cryptography and cryptology interchangeably in English, while others (including US military practice generally) use cryptography to refer specifically to the use and practice of cryptographic techniques and cryptography to refer to the combined study of cryptography and cryptanalysis. English is more flexible than several other languages in which *cryptology* (done by cryptologists) is always used in the second sense above. [RFC 2828](https://tools.ietf.org/html/rfc2828) advises that[steganography](https://en.wikipedia.org/wiki/Steganography) is sometimes included in cryptology.

The study of characteristics of languages that have some application in cryptography or cryptology (e.g. frequency data, letter combinations, universal patterns, etc.) is called cryptolinguistics.

**HISTORY OF CRYPTOGRAPHY**

Before the modern era, cryptography focused on message confidentiality (i.e., encryption)—conversion of [messages](https://en.wikipedia.org/wiki/Information) from a comprehensible form into an incomprehensible one and back again at the other end, rendering it unreadable by interceptors or eavesdroppers without secret knowledge (namely the key needed for decryption of that message). Encryption attempted to ensure [secrecy](https://en.wikipedia.org/wiki/Secrecy) in [communications](https://en.wikipedia.org/wiki/Communications), such as those of [spies](https://en.wikipedia.org/wiki/Spy), military leaders, and [diplomats](https://en.wikipedia.org/wiki/Diplomat). In recent decades, the field has expanded beyond confidentiality concerns to include techniques for message integrity checking, sender/receiver identity [authentication](https://en.wikipedia.org/wiki/Authentication), [digital signatures](https://en.wikipedia.org/wiki/Digital_signature), [interactive proofs](https://en.wikipedia.org/wiki/Interactive_proof_system) and [secure computation](https://en.wikipedia.org/wiki/Secure_multiparty_computation), among others.

Cryptography, the use of codes and ciphers to protect secrets, began thousands of years ago. Until recent decades, it has been the story of what might be called [classic cryptography](https://en.wikipedia.org/wiki/Classical_cryptography) — that is, of methods of [encryption](https://en.wikipedia.org/wiki/Encryption) that use pen and paper, or perhaps simple mechanical aids. In the early 20th century, the invention of complex mechanical and electromechanical machines, such as the [Enigma](https://en.wikipedia.org/wiki/Enigma_(machine)) [rotor machine](https://en.wikipedia.org/wiki/Rotor_machine), provided more sophisticated and efficient means of encryption; and the subsequent introduction of electronics and computing has allowed elaborate schemes of still greater complexity, most of which are entirely unsuited to pen and paper.

The development of [cryptography](https://en.wikipedia.org/wiki/Cryptography) has been paralleled by the development of [cryptanalysis](https://en.wikipedia.org/wiki/Cryptanalysis) — the "breaking" of codes and[ciphers](https://en.wikipedia.org/wiki/Cipher). The discovery and application, early on, of [frequency analysis](https://en.wikipedia.org/wiki/Frequency_analysis) to the reading of encrypted communications has, on occasion, altered the course of history. Thus the [Zimmermann Telegram](https://en.wikipedia.org/wiki/Zimmermann_Telegram) triggered the United States' entry into World War I; and [Allied](https://en.wikipedia.org/wiki/Allies_of_World_War_II) reading of [Nazi Germany](https://en.wikipedia.org/wiki/Nazi_Germany)'s ciphers shortened World War II, in some evaluations by as much as two years.

Until the 1970s, secure cryptography was largely the preserve of governments. Two events have since brought it squarely into the public domain: the creation of a public encryption standard ([DES](https://en.wikipedia.org/wiki/Data_Encryption_Standard)), and the invention of [public-key cryptography](https://en.wikipedia.org/wiki/Public-key_cryptography).

**CLASSICAL CRYPTOGRAPHY**

The earliest known use of cryptography is found in non-standard [hieroglyphs](https://en.wikipedia.org/wiki/Egyptian_hieroglyphs) carved into the wall of a tomb from the [Old Kingdom of Egypt](https://en.wikipedia.org/wiki/Old_Kingdom_of_Egypt) circa 1900 BCE. These are not thought to be serious attempts at secret communications, however, but rather to have been attempts at mystery, intrigue, or even amusement for literate onlookers. These are examples of still other uses of cryptography, or of something that looks (impressively if misleadingly) like it. Some [clay tablets](https://en.wikipedia.org/wiki/Clay_tablet) from Mesopotamia somewhat later are clearly meant to protect information—one dated near 1500 BCE was found to encrypt a craftsman's recipe for pottery glaze, presumably commercially valuable. Later still, [Hebrew](https://en.wikipedia.org/wiki/Hebrew_language) scholars made use of simple monoalphabetic [substitution ciphers](https://en.wikipedia.org/wiki/Substitution_ciphers) (such as the [Atbash cipher](https://en.wikipedia.org/wiki/Atbash_cipher" \o "Atbash cipher)) beginning perhaps around 500 to 600 BCE

The [ancient Greeks](https://en.wikipedia.org/wiki/Ancient_Greeks) are said to have known of ciphers. The [scytale](https://en.wikipedia.org/wiki/Scytale" \o "Scytale) [transposition cipher](https://en.wikipedia.org/wiki/Transposition_cipher)was used by the [Spartan](https://en.wikipedia.org/wiki/Sparta) military, however it is disputed whether the scytale was for encryption, authentication, or avoiding bad omens in speech. [Herodotus](https://en.wikipedia.org/wiki/Herodotus) tells us of secret messages physically concealed beneath wax on wooden tablets or as a tattoo on a slave's head concealed by regrown hair, though these are not properly examples of cryptography *per se* as the message, once known, is directly readable; this is known as[steganography](https://en.wikipedia.org/wiki/Steganography). Another Greek method was developed by [Polybius](https://en.wikipedia.org/wiki/Polybius) (now called the "[Polybius Square](https://en.wikipedia.org/wiki/Polybius#Cryptography)"). The [Romans](https://en.wikipedia.org/wiki/Ancient_Rome) knew something of cryptography (e.g., the [Caesar cipher](https://en.wikipedia.org/wiki/Caesar_cipher) and its variations).

## MEDIEVAL AND RENAISSANCE CRYPTOGRAPHY

The first page of [al-Kindi](https://en.wikipedia.org/wiki/Al-Kindi)'s manuscript on deciphering cryptographic messages containing the first descriptions of cryptanalysis and frequency analysis.

[David Kahn](https://en.wikipedia.org/wiki/David_Kahn_(writer)) notes in codebreakers  that modern cryptology originated among the Arabs, the first people to systematically document cryptanalytic methods. The invention of the frequency-analysis technique for breaking monoalphabetic[substitution ciphers](https://en.wikipedia.org/wiki/Substitution_cipher), by [Al-Kindi](https://en.wikipedia.org/wiki/Al-Kindi), an [Arab mathematician](https://en.wikipedia.org/wiki/Mathematics_in_medieval_Islam), sometime around AD 800 proved to be the single most significant cryptanalytic advance until World War II. Al-Kindi wrote a book on cryptography entitled risalah fi istikhraj al-mu’amma (manuscript for the deciphering cryptographic messages in which he described the first cryptanalytic techniques, including some for [polyalphabetic ciphers](https://en.wikipedia.org/wiki/Polyalphabetic_cipher), cipher classification, Arabic phonetics and syntax, and most importantly, gave the first descriptions on frequency analysis. He also covered methods of encipherments, cryptanalysis of certain encipherments, and statistical analysis of letters and letter combinations in Arabic.

[Ahmad al-Qalqashandi](https://en.wikipedia.org/wiki/Ahmad_al-Qalqashandi) (AD 1355–1418) wrote the subh al-a ‘sha, a 14-volume encyclopedia which included a section on cryptology. This information was attributed to [Ibn al-Durayhim](https://en.wikipedia.org/wiki/Ibn_al-Durayhim" \o "Ibn al-Durayhim) who lived from AD 1312 to 1361, but whose writings on cryptography have been lost. The list of ciphers in this work included both [substitution](https://en.wikipedia.org/wiki/Substitution_cipher) and [transposition](https://en.wikipedia.org/wiki/Transposition_cipher), and for the first time, a cipher with multiple substitutions for each [plaintext](https://en.wikipedia.org/wiki/Plaintext) letter. Also traced to Ibn al-Durayhim is an exposition on and worked example of cryptanalysis, including the use of tables of [letter frequencies](https://en.wikipedia.org/wiki/Letter_frequencies) and sets of letters which cannot occur together in one word.

The earliest example of the homophonic [substitution cipher](https://en.wikipedia.org/wiki/Substitution_cipher) is the one used by [Duke of Mantua](https://en.wikipedia.org/wiki/Duke_of_Mantua) in the early 1400s. Homophonic cipher replaces each letter with multiple symbols depending on the letter frequency. The cipher is ahead of the time because it combines monoalphabetic and polyalphabetic features.

Essentially all ciphers remained vulnerable to the cryptanalytic technique of frequency analysis until the development of the polyalphabetic cipher, and many remained so thereafter. The polyalphabetic cipher was most clearly explained by [Leon Battista Alberti](https://en.wikipedia.org/wiki/Leon_Battista_Alberti) around the year AD 1467, for which he was called the "father of Western cryptology". [Johannes Trithemius](https://en.wikipedia.org/wiki/Johannes_Trithemius), in his work poligraphia, invented the [tabula recta](https://en.wikipedia.org/wiki/Tabula_recta), a critical component of the Vigenère cipher. The French cryptographer[Blaise de Vigenère](https://en.wikipedia.org/wiki/Blaise_de_Vigen%C3%A8re) devised a practical polyalphabetic system which bears his name, the [Vigenère cipher](https://en.wikipedia.org/wiki/Vigen%C3%A8re_cipher" \o "Vigenère cipher).

In Europe, cryptography became (secretly) more important as a consequence of political competition and religious revolution. For instance, in Europe during and after the [Renaissance](https://en.wikipedia.org/wiki/Renaissance), citizens of the various Italian states—the [Papal States](https://en.wikipedia.org/wiki/Papal_States)and the Roman Catholic Church included—were responsible for rapid proliferation of cryptographic techniques, few of which reflect understanding (or even knowledge) of Alberti's polyalphabetic advance. 'Advanced ciphers', even after Alberti, weren't as advanced as their inventors / developers / users claimed (and probably even themselves believed). They were regularly broken. This over-optimism may be inherent in cryptography, for it was then - and remains today - fundamentally difficult to accurately know how vulnerable one's system actually is. In the absence of knowledge, guesses and hopes, predictably, are common.

Cryptography, [cryptanalysis](https://en.wikipedia.org/wiki/Cryptanalysis), and secret-agent/courier betrayal featured in the [Babington plot](https://en.wikipedia.org/wiki/Babington_plot) during the reign of Queen[Elizabeth I](https://en.wikipedia.org/wiki/Elizabeth_I_of_England) which led to the execution of [Mary, Queen of Scots](https://en.wikipedia.org/wiki/Mary,_Queen_of_Scots).

The chief cryptographer of King Louis XIV of France was Antoine Rossignol and he and his family created what is known as the [Great Cipher](https://en.wikipedia.org/wiki/Great_Cipher) because it remained unsolved from its initial use until 1890, when French military cryptanalyst, [Étienne Bazeries](https://en.wikipedia.org/wiki/%C3%89tienne_Bazeries) solved it.[[13]](https://en.wikipedia.org/wiki/History_of_cryptography#cite_note-13) An encrypted message from the time of the [Man in the Iron Mask](https://en.wikipedia.org/wiki/Man_in_the_Iron_Mask) (decrypted just prior to 1900 by[Étienne Bazeries](https://en.wikipedia.org/wiki/%C3%89tienne_Bazeries)) has shed some, regrettably non-definitive, light on the identity of that real, if legendary and unfortunate, prisoner.

Outside of Europe, after the Mongols brought about the end of the Muslim Golden Age, cryptography remained comparatively undeveloped. [Cryptography in Japan](https://en.wikipedia.org/wiki/Cryptography_in_Japan) seems not to have been used until about 1510, and advanced techniques were not known until after the opening of the country to the West beginning in the 1860s.

## CRYPTOGRAPHY FROM 1800 TO WORLD WAR II

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

Cryptography, and its misuse, were involved in the execution of [Mata Hari](https://en.wikipedia.org/wiki/Mata_Hari) and in [Dreyfus' conviction](https://en.wikipedia.org/wiki/Dreyfus_affair) and imprisonment, both in the early 20th century. Cryptographers were also involved in exposing the machinations which had led to the Dreyfus affair; Mata Hari, in contrast, was shot.

In World War I the [Admiralty](https://en.wikipedia.org/wiki/Admiralty)'s [Room 40](https://en.wikipedia.org/wiki/Room_40) broke German naval codes and played an important role in several naval engagements during the war, notably in detecting major German sorties into the [North Sea](https://en.wikipedia.org/wiki/North_Sea) that led to the battles of [Dogger Bank](https://en.wikipedia.org/wiki/Battle_of_Dogger_Bank_(1915)) and [Jutland](https://en.wikipedia.org/wiki/Battle_of_Jutland) as the British fleet was sent out to intercept them. However its most important contribution was probably in[decrypting](https://en.wikipedia.org/wiki/Cryptanalysis) the [Zimmermann Telegram](https://en.wikipedia.org/wiki/Zimmermann_Telegram), a [cable](https://en.wikipedia.org/wiki/Telegram) from the German Foreign Office sent via Washington to its [ambassador](https://en.wikipedia.org/wiki/Ambassador" \o "Ambassador)[Heinrich von Eckardt](https://en.wikipedia.org/wiki/Heinrich_von_Eckardt) in Mexico which played a major part in bringing the United States into the war.

In 1917, [Gilbert Vernam](https://en.wikipedia.org/wiki/Gilbert_Vernam) proposed a teleprinter cipher in which a previously prepared key, kept on paper tape, is combined character by character with the plaintext message to produce the cyphertext. This led to the development of electromechanical devices as cipher machines, and to the only unbreakable cipher, the [one time pad](https://en.wikipedia.org/wiki/One_time_pad" \o "One time pad).

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

Mathematical methods proliferated in the period prior to World War II (notably in [William F. Friedman](https://en.wikipedia.org/wiki/William_F._Friedman)'s application of statistical techniques to cryptanalysis and cipher development and in [Marian Rejewski](https://en.wikipedia.org/wiki/Marian_Rejewski)'s initial break into the German Army's version of the [Enigma](https://en.wikipedia.org/wiki/Enigma_(machine)) system in 1932).

**WORLD WAR 1**

* **World War I** (**WWI** or **WW1**), also known as the **First World War**, the**Great War**, or the **War to End All Wars**, was a [global war](https://en.wikipedia.org/wiki/World_war) originating in Europe that lasted from 28 July 1914 to 11 November 1918. More than 70 million [military personnel](https://en.wikipedia.org/wiki/Military_personnel), including 60 million Europeans, were mobilised in one of the largest wars in history. Over nine million[combatants](https://en.wikipedia.org/wiki/Combatants) and seven million [civilians](https://en.wikipedia.org/wiki/Civilian) [died as a result of the war](https://en.wikipedia.org/wiki/World_War_I_casualties)(including the victims of a [number of genocides](https://en.wikipedia.org/wiki/Genocides_in_history#World_War_I_through_World_War_II)), a casualty rate exacerbated by the belligerents' [technological and industrial sophistication](https://en.wikipedia.org/wiki/Second_Industrial_Revolution), and the tactical stalemate caused by gruelling [trench warfare](https://en.wikipedia.org/wiki/Trench_warfare). It was [one of the deadliest conflicts in history](https://en.wikipedia.org/wiki/List_of_wars_and_anthropogenic_disasters_by_death_toll), and paved the way for major political changes, including revolutions in many of the nations involved and to the [Second World War](https://en.wikipedia.org/wiki/World_War_II) twenty-one years later.
* The war drew in all the world's economic [great powers](https://en.wikipedia.org/wiki/Great_power), assembled in two opposing alliances: the [Allies](https://en.wikipedia.org/wiki/Allies_of_World_War_I) (based on the [Triple Entente](https://en.wikipedia.org/wiki/Triple_Entente) of the[Russian Empire](https://en.wikipedia.org/wiki/Russian_Empire), the [French Third Republic](https://en.wikipedia.org/wiki/French_Third_Republic), and the [United Kingdom of Great Britain and Ireland](https://en.wikipedia.org/wiki/United_Kingdom_of_Great_Britain_and_Ireland)) versus the [Central Powers](https://en.wikipedia.org/wiki/Central_Powers) of [Germany](https://en.wikipedia.org/wiki/German_Empire) and[Austria-Hungary](https://en.wikipedia.org/wiki/Austria-Hungary). Although [Italy](https://en.wikipedia.org/wiki/Kingdom_of_Italy) was a member of the [Triple Alliance](https://en.wikipedia.org/wiki/Triple_Alliance_(1882))alongside Germany and Austria-Hungary, it did not join the Central Powers, as Austria-Hungary had taken the offensive against the terms of the alliance.[[9]](https://en.wikipedia.org/wiki/World_War_I#cite_note-Willmott15-17) These alliances were reorganised and expanded as more nations entered the war: Italy, [Japan](https://en.wikipedia.org/wiki/Empire_of_Japan) and the [United States](https://en.wikipedia.org/wiki/United_States) joined the Allies, while the [Ottoman Empire](https://en.wikipedia.org/wiki/Ottoman_Empire) and [Bulgaria](https://en.wikipedia.org/wiki/Kingdom_of_Bulgaria) joined the Central Powers.
* The trigger for the war was the [assassination of Archduke Franz Ferdinand of Austria](https://en.wikipedia.org/wiki/Assassination_of_Archduke_Franz_Ferdinand_of_Austria), heir to the throne of Austria-Hungary, by[Yugoslav nationalist](https://en.wikipedia.org/wiki/Yugoslav_nationalism) [Gavrilo Princip](https://en.wikipedia.org/wiki/Gavrilo_Princip" \o "Gavrilo Princip) in [Sarajevo](https://en.wikipedia.org/wiki/Sarajevo) on 28 June 1914. This set off a [diplomatic crisis](https://en.wikipedia.org/wiki/July_Crisis) when Austria-Hungary delivered an ultimatum to the [Kingdom of Serbia](https://en.wikipedia.org/wiki/Kingdom_of_Serbia), and entangled international alliances formed over the previous decades were invoked. Within weeks, the major powers were at war and the conflict soon spread around the world.
* On 25 July Russia began mobilisation and on 28 July the Austro-Hungarians declared war on Serbia. Germany presented an ultimatum to Russia to demobilise, and when this was refused, declared war on Russia on 1 August. Being outnumbered on the [Eastern Front](https://en.wikipedia.org/wiki/Eastern_Front_(World_War_I)), Russia urged its Triple Entente ally France to open up a second front in the west. Back in 1870, the Franco-Prussian War had ended the Second French Empire and ceded the provinces of Alsace-Lorraine to a unified Germany. Bitterness over that defeat and the determination to retake Alsace-Lorraine made the acceptance of Russia's plea for help an easy choice, so France began full mobilisation on 1 August and, on 3 August, Germany declared war on France. The border between France and Germany was heavily fortified on both sides so according to the Schlieffen Plan, Germany then invaded neutral Belgium and Luxembourg before moving towards France from the north, leading the United Kingdom to declare war on Germany on 4 August due to their violation of Belgian neutrality. After the German march on Paris was halted in the Battle of the Marne, what became known as the[Western Front](https://en.wikipedia.org/wiki/Western_Front_(World_War_I)) settled into a [battle of attrition](https://en.wikipedia.org/wiki/Attrition_warfare), with a [trench line](https://en.wikipedia.org/wiki/Trench_warfare) that changed little until 1917. On the [Eastern Front](https://en.wikipedia.org/wiki/Eastern_Front_(World_War_I)), the Russian army led a successful campaign against the Austro-Hungarians, but the Germans stopped its [invasion of East Prussia](https://en.wikipedia.org/wiki/Invasion_of_East_Prussia) in the battles of Tannenberg and the Masurian Lakes. In November 1914, the Ottoman Empire joined the Central Powers, opening fronts in the [Caucasus](https://en.wikipedia.org/wiki/Caucasus), [Mesopotamia](https://en.wikipedia.org/wiki/Mesopotamia) and the[Sinai](https://en.wikipedia.org/wiki/Sinai). In 1915, Italy joined the Allies and Bulgaria joined the Central Powers; [Romania](https://en.wikipedia.org/wiki/Kingdom_of_Romania) joined the Allies in 1916, as did the United States in 1917.
* The Russian government [collapsed in March 1917](https://en.wikipedia.org/wiki/February_Revolution), and [a revolution in November](https://en.wikipedia.org/wiki/October_Revolution) followed by a further military defeat brought the Russians to terms with the Central Powers via the [Treaty of Brest Litovsk](https://en.wikipedia.org/wiki/Treaty_of_Brest_Litovsk), which granted the Germans a significant victory. After a [stunning German offensive](https://en.wikipedia.org/wiki/Spring_Offensive) along the Western Front in the spring of 1918, the Allies rallied and drove back the Germans in a [series of successful offensives](https://en.wikipedia.org/wiki/Hundred_Days_Offensive). On [4 November 1918](https://en.wikipedia.org/wiki/Armistice_of_Villa_Giusti), the Austro-Hungarian empire agreed to an armistice, and Germany, which had [its own trouble with revolutionaries](https://en.wikipedia.org/wiki/German_Revolution_of_1918%E2%80%9319), agreed to an armistice on 11 November 1918, ending the war in victory for the Allies.
* By the end of the war or soon after, the German Empire, Russian Empire, [Austro-Hungarian Empire](https://en.wikipedia.org/wiki/Austro-Hungarian_Empire) and the Ottoman Empire ceased to exist. National borders were redrawn, with several independent nations restored or created, and [Germany's colonies](https://en.wikipedia.org/wiki/German_colonial_empire) were parceled out among the victors. During the [Paris Peace Conference of 1919](https://en.wikipedia.org/wiki/Paris_Peace_Conference,_1919), the [Big Four](https://en.wikipedia.org/wiki/The_Big_Four_(World_War_I))(Britain, France, the United States and Italy) imposed their terms in a series of treaties. The [League of Nations](https://en.wikipedia.org/wiki/League_of_Nations) was formed with the aim of preventing any repetition of such a conflict. This effort failed, and economic depression, renewed nationalism, weakened successor states, and feelings of humiliation (particularly in Germany) eventually contributed to the start of [World War II](https://en.wikipedia.org/wiki/World_War_II).
* was used extensively during [World War II](https://en.wikipedia.org/wiki/World_War_II), with a plethora of [code](https://en.wikipedia.org/wiki/Code_(cryptography)) and [cipher](https://en.wikipedia.org/wiki/Cipher) systems fielded by the nations involved. In addition, the theoretical and practical aspects of cryptanalysis or code breaking was much advanced.
* Probably the most important codebreaking event of the war was the successful decryption by the Allies of the German["Enigma" Cipher](https://en.wikipedia.org/wiki/Cryptanalysis_of_the_Enigma). The first complete [break](https://en.wikipedia.org/wiki/Cryptanalysis) into Enigma was accomplished by [Poland](https://en.wikipedia.org/wiki/Cryptanalysis_of_the_Enigma) around 1932; the techniques and insights used were passed to the French and British Allies just before the outbreak of the War in 1939. They were substantially improved by British efforts at the [Bletchley Park](https://en.wikipedia.org/wiki/Bletchley_Park) research station during the War. Decryption of the [Enigma Cipher](https://en.wikipedia.org/wiki/Cryptanalysis_of_the_Enigma) allowed the Allies to read important parts of German radio traffic on important networks and was an invaluable source of [military intelligence](https://en.wikipedia.org/wiki/Military_intelligence) throughout the War. Intelligence from this source (and other high level sources, including the [Fish](https://en.wikipedia.org/wiki/Fish_(cryptography)" \o "Fish (cryptography))ciphers) was eventually called [Ultra](https://en.wikipedia.org/wiki/Ultra_(cryptography)).
* A similar break into an important Japanese cipher ([PURPLE](https://en.wikipedia.org/wiki/PURPLE)) by the US Army Signals Intelligence Service started before the US entered the War. Product from this source was called [MAGIC](https://en.wikipedia.org/wiki/Magic_(cryptography)). It was the highest security Japanese diplomatic cipher.

Keeping secrets is not easy, people are always eager to spread secrets. In early days of computing(1950-1960),there was not a great deal of emphasis on security, because systems were then closed. The protocols used for computer to computer communication were unknown to general public. So, the chance of accessing others’ information was limited. As the microcomputers evolved in 1970-1980, the issue of information security started to gain more prominence. This continued till the early 1990s.It was the internet that changed the total scenario and brought a tremendous change in the field of security. It opened up enormous opportunities for new threats and possible attacks on information. As the technologists found new ways to thwart attacks, the attackers also found new ways to beat the technologists. That is why proper security during information exchange has become a crucial issue leading to the growth of several encryption & decryption techniques ..i.e. cryptography*.*Cryptography is the art and science of achieving security by encoding messages to make them non-readable. Cyptanalysis is the technique of decoding messages from a non-readable format to readable format without knowing that how it was initially converted. Cryptology is the combination of these two.

**WORLD WAR II CRYPTOGRAPHY**

The [Enigma machine](https://en.wikipedia.org/wiki/Enigma_machine) was widely used by Nazi Germany; its cryptanalysis by the Allies provided vital [Ultra](https://en.wikipedia.org/wiki/Ultra_(cryptography)" \o "Ultra (cryptography))intelligence.

By World War II, mechanical and electromechanical [cipher machines](https://en.wikipedia.org/wiki/Cipher) were in wide use, although—where such machines were impractical—manual systems continued in use. Great advances were made in both cipher design and [cryptanalysis](https://en.wikipedia.org/wiki/Cryptanalysis), all in secrecy. Information about this period has begun to be declassified as the official British 50-year secrecy period has come to an end, as US archives have slowly opened, and as assorted memoirs and articles have appeared.

The Germans made heavy use, in several variants, of an electromechanical [rotor machine](https://en.wikipedia.org/wiki/Rotor_machine) known as [Enigma](https://en.wikipedia.org/wiki/Enigma_machine). Mathematician [Marian Rejewski](https://en.wikipedia.org/wiki/Marian_Rejewski), at Poland's [Cipher Bureau](https://en.wikipedia.org/wiki/Biuro_Szyfr%C3%B3w), in December 1932 deduced the detailed structure of the German Army Enigma, using mathematics and limited documentation supplied by Captain [Gustave Bertrand](https://en.wikipedia.org/wiki/Gustave_Bertrand" \o "Gustave Bertrand) of French [military intelligence](https://en.wikipedia.org/wiki/Military_intelligence). This was the greatest breakthrough in cryptanalysis in a thousand years and more, according to historian [David Kahn](https://en.wikipedia.org/wiki/David_Kahn_(writer)). Rejewski and his mathematical Cipher Bureau colleagues, [Jerzy Różycki](https://en.wikipedia.org/wiki/Jerzy_R%C3%B3%C5%BCycki) and [Henryk Zygalski](https://en.wikipedia.org/wiki/Henryk_Zygalski" \o "Henryk Zygalski), continued reading Enigma and keeping pace with the evolution of the German Army machine's components and encipherment procedures. As the Poles' resources became strained by the changes being introduced by the Germans, and as war loomed, the [Cipher Bureau](https://en.wikipedia.org/wiki/Biuro_Szyfr%C3%B3w), on the Polish[General Staff](https://en.wikipedia.org/wiki/General_Staff)'s instructions, on 25 July 1939, at [Warsaw](https://en.wikipedia.org/wiki/Warsaw), initiated French and British intelligence representatives into the secrets of Enigma decryption.

Soon after the [Invasion of Poland](https://en.wikipedia.org/wiki/Invasion_of_Poland) by Germany on 1 September 1939, key [Cipher Bureau](https://en.wikipedia.org/wiki/Biuro_Szyfr%C3%B3w) personnel were evacuated southeastward; on 17 September, as the [Soviet Union attacked Poland](https://en.wikipedia.org/wiki/Soviet_invasion_of_Poland) from the East, they crossed into [Romania](https://en.wikipedia.org/wiki/Romania). From there they reached Paris, France; at [PC Bruno](https://en.wikipedia.org/wiki/PC_Bruno), near Paris, they continued breaking Enigma, collaborating with British[cryptologists](https://en.wikipedia.org/wiki/Cryptologist) at [Bletchley Park](https://en.wikipedia.org/wiki/Bletchley_Park) as the British got up to speed on breaking Enigma. In due course, the British cryptographers - whose ranks included many chess masters and mathematics dons such as [Gordon Welchman](https://en.wikipedia.org/wiki/Gordon_Welchman), [Max Newman](https://en.wikipedia.org/wiki/Max_Newman), and [Alan Turing](https://en.wikipedia.org/wiki/Alan_Turing) (the conceptual founder of modern [computing](https://en.wikipedia.org/wiki/Computer)) - substantially advanced the scale and technology of [Enigma](https://en.wikipedia.org/wiki/Enigma_machine) [decryption](https://en.wikipedia.org/wiki/Decryption).

[German code breaking in World War II](https://en.wikipedia.org/wiki/German_code_breaking_in_World_War_II) also had some success, most importantly by breaking the Naval Cypher No. 3. This enabled them to track and sink Atlantic convoys. It was only [Ultra](https://en.wikipedia.org/wiki/Ultra) intelligence that finally persuaded the admiralty to change their codes in June 1943. This is surprising given the success of the British [Room 40](https://en.wikipedia.org/wiki/Room_40) code breakers in the previous world war.

At the end of the War, on 19 April 1945, Britain's top military officers were told that they could never reveal that the German Enigma cipher had been broken because it would give the defeated enemy the chance to say they "were not well and fairly beaten".

[US Navy](https://en.wikipedia.org/wiki/US_Navy) cryptographers (with cooperation from British and Dutch cryptographers after 1940) broke into several [Japanese Navy](https://en.wikipedia.org/wiki/Imperial_Japanese_Navy) crypto systems. The break into one of them, [JN-25](https://en.wikipedia.org/wiki/JN-25), famously led to the US victory in the [Battle of Midway](https://en.wikipedia.org/wiki/Battle_of_Midway); and to the publication of that fact in the [Chicago Tribune](https://en.wikipedia.org/wiki/Chicago_Tribune) shortly after the battle, though the Japanese seem not to have noticed for they kept using the JN-25 system. A US Army group, the [SIS](https://en.wikipedia.org/wiki/Signals_Intelligence_Service), managed to break the highest security Japanese diplomatic cipher system (an electromechanical 'stepping switch' machine called [Purple](https://en.wikipedia.org/wiki/Purple_code) by the Americans) even before World War II began. The Americans referred to the intelligence resulting from cryptanalysis, perhaps especially that from the Purple machine, as '[Magic](https://en.wikipedia.org/wiki/Magic_cryptography)'. The British eventually settled on '[Ultra](https://en.wikipedia.org/wiki/Ultra_(cryptography))' for intelligence resulting from cryptanalysis, particularly that from message traffic protected by the various Enigmas. An earlier British term for Ultra had been 'Boniface' in an attempt to suggest, if betrayed, that it might have an individual agent as a source.

The German military also deployed several mechanical attempts at a [one-time pad](https://en.wikipedia.org/wiki/One-time_pad). Bletchley Park called them the [Fish ciphers](https://en.wikipedia.org/wiki/FISH_(cryptography)), and [Max Newman](https://en.wikipedia.org/wiki/Max_Newman) and colleagues designed and deployed the [Heath Robinson](https://en.wikipedia.org/wiki/Heath_Robinson_(codebreaking_machine)), and then the world's first programmable digital electronic computer, the [Colossus](https://en.wikipedia.org/wiki/Colossus_computer), to help with their cryptanalysis. The German Foreign Office began to use the [one-time pad](https://en.wikipedia.org/wiki/One-time_pad) in 1919; some of this traffic was read in World War II partly as the result of recovery of some key material in South America that was discarded without sufficient care by a German courier.

The Japanese Foreign Office used a locally developed electrical stepping switch based system (called [Purple](https://en.wikipedia.org/wiki/Purple_code) by the US), and also had used several similar machines for attaches in some Japanese embassies. One of the electrical stepping switch based systems referred to earlier as [Purple](https://en.wikipedia.org/wiki/Purple_code) was called the 'M-machine' by the US, another was referred to as 'Red'. All were broken, to one degree or another, by the Allies.

[Allied](https://en.wikipedia.org/wiki/Allies_of_World_War_II) cipher machines used in World War II included the British[TypeX](https://en.wikipedia.org/wiki/TypeX) and the American [SIGABA](https://en.wikipedia.org/wiki/SIGABA); both were electromechanical rotor designs similar in spirit to the Enigma, albeit with major improvements. Neither is known to have been broken by anyone during the War. The Poles used the [Lacida](https://en.wikipedia.org/wiki/Lacida" \o "Lacida) machine, but its security was found to be less than intended (by Polish Army cryptographers in the UK), and its use was discontinued. US troops in the field used the [M-209](https://en.wikipedia.org/wiki/M-209) and the still less secure [M-94](https://en.wikipedia.org/wiki/M-94) family machines. British[SOE](https://en.wikipedia.org/wiki/Special_Operations_Executive) agents initially used 'poem ciphers' (memorized poems were the encryption/decryption keys), but later in the War, they began to[switch](https://en.wikipedia.org/wiki/Leo_Marks) to [one-time pads](https://en.wikipedia.org/wiki/One-time_pad).

The [VIC cipher](https://en.wikipedia.org/wiki/VIC_cipher) (used at least until 1957 in connection with [Rudolf Abel](https://en.wikipedia.org/wiki/Rudolf_Abel)'s NY spy ring) was a very complex hand cipher, and is claimed to be the most complicated known to have been used by the Soviets, according to David Kahn in kahn on codes. For the decrypting of Soviet ciphers (particularly when one time pads were reused), see [Venona project](https://en.wikipedia.org/wiki/Venona_project" \o "Venona project).

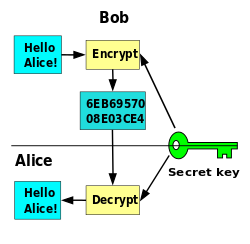
## MODERN CRYPTOGRAPHY

Encryption in modern times is achieved by using algorithms that have a key to encrypt and decrypt information. These keys convert the messages and data into “digital gibberish” through encryption and then return them to the original form through decryption. In general, the longer the key is, the more difficult it is to crack the code. This holds true because deciphering an encrypted message by brute force would require the attacker to try every possible key. To put this in context, each binary unit of information, or bit, has a value of 0 or 1. An 8-bit key would then have 256 or 2^8 possible keys. A 56-bit key would have 2^56, or 72 quadrillion, possible keys to try and decipher the message. With modern technology, cyphers using keys with these lengths are becoming easier to decipher. DES, an early US Government approved cypher, has an effective key length of 56 bits, and test messages using that cypher have been broken by brute force key search. However, as technology advances, so does the quality of encryption. Since World War II, one of the most notable advances in the study of cryptography is the introduction of the asymmetric key cyphers (sometimes termed public-key cyphers). These are algorithms which use two mathematically related keys for encryption of the same message. Some of these algorithms permit publication of one of the keys, due to it being extremely difficult to determine one key simply from knowledge of the other.

Beginning around 1990, the use of the [Internet](https://en.wikipedia.org/wiki/Internet) for commercial purposes and the introduction of commercial transactions over the Internet called for a widespread standard for encryption. Before the introduction of the Advanced Encryption Standard (AES), information sent over the Internet, such as financial data, was encrypted if at all, most commonly using the Data Encryption Standard (DES). This had been approved by NBS (a US Government agency) for its security, after public call for, and a comptetition among, candidates for such a cypher algorithm. DES was approved for a short period, but saw extended use due to complex wrangles over the use by the public of high quality encryption. DES was finally replaced by the AES after another public competition organized by the NBS successor agency, NIST. Around the late 1990s to early 2000s, the use of public-key algorithms became a more common approach for encryption, and soon a [hybrid of the two schemes](https://en.wikipedia.org/wiki/Hybrid_cryptosystem)became the most accepted way for e-commerce operations to proceed. Additionally, the creation of a new protocol known as the Secure Socket Layer, or SSL, led the way for online transactions to take place. Transactions ranging from purchasing goods to online bill pay and banking used SSL. Furthermore, as wireless Internet connections became more common among households, the need for encryption grew, as a level of security was needed in these everyday situations.

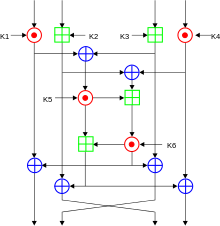
The modern field of cryptography can be divided into several areas of study. The chief ones are discussed here; see [Topics in Cryptography](https://en.wikipedia.org/wiki/Topics_in_Cryptography) for more.

### SYMMETRIC-KEY CRYPTOGRAPHY.

[](https://en.wikipedia.org/wiki/File:Symmetric_key_encryption.svg)

Symmetric-key cryptography, where a single key is used for encryption and decryption

Symmetric-key cryptography refers to encryption methods in which both the sender and receiver share the same key (or, less commonly, in which their keys are different, but related in an easily computable way). This was the only kind of encryption publicly known until June 1976.

[](https://en.wikipedia.org/wiki/File:International_Data_Encryption_Algorithm_InfoBox_Diagram.svg)

One round (out of 8.5) of the [IDEA](https://en.wikipedia.org/wiki/International_Data_Encryption_Algorithm" \o "International Data Encryption Algorithm)cipher, used in some versions of [PGP](https://en.wikipedia.org/wiki/Pretty_Good_Privacy" \o "Pretty Good Privacy)for high-speed encryption of, for instance, [e-mail](https://en.wikipedia.org/wiki/Electronic_mail)

Symmetric key ciphers are implemented as either [block ciphers](https://en.wikipedia.org/wiki/Block_ciphers) or [stream ciphers](https://en.wikipedia.org/wiki/Stream_ciphers). A block cipher enciphers input in blocks of plaintext as opposed to individual characters, the input form used by a stream cipher.

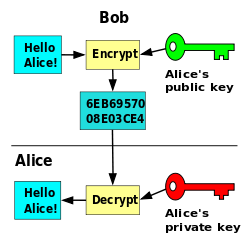
The [Data Encryption Standard](https://en.wikipedia.org/wiki/Data_Encryption_Standard) (DES) and the [Advanced Encryption Standard](https://en.wikipedia.org/wiki/Advanced_Encryption_Standard)(AES) are block cipher designs that have been designated [cryptography standards](https://en.wikipedia.org/wiki/Cryptography_standards) by the US government (though DES's designation was finally withdrawn after the AES was adopted). Despite its deprecation as an official standard, DES (especially its still-approved and much more secure [triple-DES](https://en.wikipedia.org/wiki/Triple-DES)variant) remains quite popular; it is used across a wide range of applications, from ATM encryption to [e-mail privacy](https://en.wikipedia.org/wiki/E-mail_privacy) and [secure remote access](https://en.wikipedia.org/wiki/Secure_Shell). Many other block ciphers have been designed and released, with considerable variation in quality. Many have been thoroughly broken, such as [FEAL](https://en.wikipedia.org/wiki/FEAL).

Stream ciphers, in contrast to the 'block' type, create an arbitrarily long stream of key material, which is combined with the plaintext bit-by-bit or character-by-character, somewhat like the [one-time pad](https://en.wikipedia.org/wiki/One-time_pad). In a stream cipher, the output stream is created based on a hidden internal state that changes as the cipher operates. That internal state is initially set up using the secret key material. [RC4](https://en.wikipedia.org/wiki/RC4) is a widely used stream cipher; see [Category:Stream ciphers](https://en.wikipedia.org/wiki/Category:Stream_ciphers" \o "Category:Stream ciphers).[[4]](https://en.wikipedia.org/wiki/Cryptography#cite_note-hac-4) Block ciphers can be used as stream ciphers; see [Block cipher modes of operation](https://en.wikipedia.org/wiki/Block_cipher_modes_of_operation).

[Cryptographic hash functions](https://en.wikipedia.org/wiki/Cryptographic_hash_functions) are a third type of cryptographic algorithm. They take a message of any length as input, and output a short, fixed length [hash](https://en.wikipedia.org/wiki/Hash_function), which can be used in (for example) a digital signature. For good hash functions, an attacker cannot find two messages that produce the same hash. [MD4](https://en.wikipedia.org/wiki/MD4) is a long-used hash function that is now broken; [MD5](https://en.wikipedia.org/wiki/MD5), a strengthened variant of MD4, is also widely used but broken in practice. The US [National Security Agency](https://en.wikipedia.org/wiki/National_Security_Agency) developed the Secure Hash Algorithm series of MD5-like hash functions: SHA-0 was a flawed algorithm that the agency withdrew; [SHA-1](https://en.wikipedia.org/wiki/SHA-1) is widely deployed and more secure than MD5, but cryptanalysts have identified attacks against it; the [SHA-2](https://en.wikipedia.org/wiki/SHA-2) family improves on SHA-1, but it isn't yet widely deployed; and the US standards authority thought it "prudent" from a security perspective to develop a new standard to "significantly improve the robustness of NIST's overall hash algorithm toolkit." Thus, a [hash function design competition](https://en.wikipedia.org/wiki/NIST_hash_function_competition) was meant to select a new U.S. national standard, to be called [SHA-3](https://en.wikipedia.org/wiki/SHA-3), by 2012. The competition ended on October 2, 2012 when the NIST announced that [Keccak](https://en.wikipedia.org/wiki/Keccak" \o "Keccak) would be the new SHA-3 hash algorithm. Unlike block and stream ciphers that are invertible, cryptographic hash functions produce a hashed output that cannot be used to retrieve the original input data. Cryptographic hash functions are used to verify the authenticity of data retrieved from an untrusted source or to add a layer of security.

[Message authentication codes](https://en.wikipedia.org/wiki/Message_authentication_code) (MACs) are much like cryptographic hash functions, except that a secret key can be used to authenticate the hash value upon receipt; this additional complication blocks an attack scheme against bare digest algorithms, and so has been thought worth the effort.

### PUBLIC-KEY CRYPTOGRAPHY

[](https://en.wikipedia.org/wiki/File:Public_key_encryption.svg)

Public-key cryptography, where different keys are used for encryption and decryption

Symmetric-key cryptosystems use the same key for encryption and decryption of a message, though a message or group of messages may have a different key than others. A significant disadvantage of symmetric ciphers is the [key management](https://en.wikipedia.org/wiki/Key_management) necessary to use them securely. Each distinct pair of communicating parties must, ideally, share a different key, and perhaps each ciphertext exchanged as well. The number of keys required increases as the[square](https://en.wikipedia.org/wiki/Square_(algebra)) of the number of network members, which very quickly requires complex key management schemes to keep them all consistent and secret. The difficulty of securely establishing a secret key between two communicating parties, when a [secure channel](https://en.wikipedia.org/wiki/Secure_channel) does not already exist between them, also presents a[chicken-and-egg problem](https://en.wikipedia.org/wiki/Chicken-and-egg_problem) which is a considerable practical obstacle for cryptography users in the real world.

[Whitfield Diffie](https://en.wikipedia.org/wiki/Whitfield_Diffie) and [Martin Hellman](https://en.wikipedia.org/wiki/Martin_Hellman), authors of the first published paper on public-key cryptography

In a groundbreaking 1976 paper, Whitfield Diffie and Martin Hellman proposed the notion of public key (also, more generally, called asymmetric key) cryptography in which two different but mathematically related keys are used—a public key and a private key. A public key system is so constructed that calculation of one key (the 'private key') is computationally infeasible from the other (the 'public key'), even though they are necessarily related. Instead, both keys are generated secretly, as an interrelated pair. The historian [David Kahn](https://en.wikipedia.org/wiki/David_Kahn_(writer)) described public-key cryptography as "the most revolutionary new concept in the field since polyalphabetic substitution emerged in the Renaissance".

In public-key cryptosystems, the public key may be freely distributed, while its paired private key must remain secret. In a public-key encryption system, the public key is used for encryption, while the *private* or *secret key* is used for decryption. While Diffie and Hellman could not find such a system, they showed that public-key cryptography was indeed possible by presenting the [Diffie–Hellman key exchange](https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange" \o "Diffie–Hellman key exchange) protocol, a solution that is now widely used in secure communications to allow two parties to secretly agree on a [shared encryption key](https://en.wikipedia.org/wiki/Symmetric-key_algorithm).

Diffie and Hellman's publication sparked widespread academic efforts in finding a practical public-key encryption system. This race was finally won in 1978 by [Ronald Rivest](https://en.wikipedia.org/wiki/Ronald_Rivest), [Adi Shamir](https://en.wikipedia.org/wiki/Adi_Shamir" \o "Adi Shamir), and [Len Adleman](https://en.wikipedia.org/wiki/Len_Adleman), whose solution has since become known as the [RSA algorithm](https://en.wikipedia.org/wiki/RSA_(algorithm)).

The Diffie–Hellman and RSA algorithms, in addition to being the first publicly known examples of high quality public-key algorithms, have been among the most widely used. Others include the [Cramer–Shoup cryptosystem](https://en.wikipedia.org/wiki/Cramer%E2%80%93Shoup_cryptosystem), [ElGamal encryption](https://en.wikipedia.org/wiki/ElGamal_encryption" \o "ElGamal encryption), and various [elliptic curve techniques](https://en.wikipedia.org/wiki/Elliptic_curve_cryptography). See [Category:Asymmetric-key cryptosystems](https://en.wikipedia.org/wiki/Category:Asymmetric-key_cryptosystems" \o "Category:Asymmetric-key cryptosystems).

To much surprise, a document published in 1997 by the Government Communications Headquarters ([GCHQ](https://en.wikipedia.org/wiki/GCHQ)), a British intelligence organization, revealed that cryptographers at GCHQ had anticipated several academic developments.[[37]](https://en.wikipedia.org/wiki/Cryptography#cite_note-nytimes-37)Reportedly, around 1970, [James H. Ellis](https://en.wikipedia.org/wiki/James_H._Ellis) had conceived the principles of asymmetric key cryptography. In 1973, [Clifford Cocks](https://en.wikipedia.org/wiki/Clifford_Cocks) invented a solution that essentially resembles the RSA algorithm. And in 1974, [Malcolm J. Williamson](https://en.wikipedia.org/wiki/Malcolm_J._Williamson) is claimed to have developed the Diffie–Hellman key exchange.

Public-key cryptography can also be used for implementing [digital signature](https://en.wikipedia.org/wiki/Digital_signature)schemes. A digital signature is reminiscent of an ordinary [signature](https://en.wikipedia.org/wiki/Signature); they both have the characteristic of being easy for a user to produce, but difficult for anyone else to[forge](https://en.wikipedia.org/wiki/Forgery). Digital signatures can also be permanently tied to the content of the message being signed; they cannot then be 'moved' from one document to another, for any attempt will be detectable. In digital signature schemes, there are two algorithms: one for *signing*, in which a secret key is used to process the message (or a hash of the message, or both), and one for *verification,* in which the matching public key is used with the message to check the validity of the signature. RSA and [DSA](https://en.wikipedia.org/wiki/Digital_Signature_Algorithm) are two of the most popular digital signature schemes. Digital signatures are central to the operation of [public key infrastructures](https://en.wikipedia.org/wiki/Public_key_infrastructure) and many network security schemes (e.g., [SSL/TLS](https://en.wikipedia.org/wiki/Transport_Layer_Security), many [VPNs](https://en.wikipedia.org/wiki/VPN), etc.).

Public-key algorithms are most often based on the [computational complexity](https://en.wikipedia.org/wiki/Computational_complexity_theory) of "hard" problems, often from [number theory](https://en.wikipedia.org/wiki/Number_theory). For example, the hardness of RSA is related to the [integer factorization](https://en.wikipedia.org/wiki/Integer_factorization) problem, while Diffie–Hellman and DSA are related to the [discrete logarithm](https://en.wikipedia.org/wiki/Discrete_logarithm) problem. More recently, [elliptic curve cryptography](https://en.wikipedia.org/wiki/Elliptic_curve_cryptography) has developed, a system in which security is based on number theoretic problems involving [elliptic curves](https://en.wikipedia.org/wiki/Elliptic_curve). Because of the difficulty of the underlying problems, most public-key algorithms involve operations such as [modular](https://en.wikipedia.org/wiki/Modular_arithmetic) multiplication and exponentiation, which are much more computationally expensive than the techniques used in most block ciphers, especially with typical key sizes. As a result, public-key cryptosystems are commonly [hybrid cryptosystems](https://en.wikipedia.org/wiki/Hybrid_cryptosystem), in which a fast high-quality symmetric-key encryption algorithm is used for the message itself, while the relevant symmetric key is sent with the message, but encrypted using a public-key algorithm. Similarly, hybrid signature schemes are often used, in which a cryptographic hash function is computed, and only the resulting hash is digitally signed.

### CRYPTANALYSIS

The goal of cryptanalysis is to find some weakness or insecurity in a cryptographic scheme, thus permitting its subversion or evasion.

It is a common misconception that every encryption method can be broken. In connection with his WWII work at [Bell Labs](https://en.wikipedia.org/wiki/Bell_Labs), [Claude Shannon](https://en.wikipedia.org/wiki/Claude_Shannon) proved that the [one-time pad](https://en.wikipedia.org/wiki/One-time_pad) cipher is unbreakable, provided the key material is truly [random](https://en.wikipedia.org/wiki/Statistical_randomness), never reused, kept secret from all possible attackers, and of equal or greater length than the message.[[40]](https://en.wikipedia.org/wiki/Cryptography#cite_note-40) Most ciphers, apart from the one-time pad, can be broken with enough computational effort by [brute force attack](https://en.wikipedia.org/wiki/Brute_force_attack), but the amount of effort needed may be [exponentially](https://en.wikipedia.org/wiki/Exponential_time) dependent on the key size, as compared to the effort needed to make use of the cipher. In such cases, effective security could be achieved if it is proven that the effort required (i.e., "work factor", in Shannon's terms) is beyond the ability of any adversary. This means it must be shown that no efficient method (as opposed to the time-consuming brute force method) can be found to break the cipher. Since no such proof has been found to date, the one-time-pad remains the only theoretically unbreakable cipher.

There are a wide variety of cryptanalytic attacks, and they can be classified in any of several ways. A common distinction turns on what Eve (an attacker) knows and what capabilities are available. In a [ciphertext-only attack](https://en.wikipedia.org/wiki/Ciphertext-only_attack" \o "Ciphertext-only attack), Eve has access only to the ciphertext (good modern cryptosystems are usually effectively immune to ciphertext-only attacks). In a [known-plaintext attack](https://en.wikipedia.org/wiki/Known-plaintext_attack), Eve has access to a ciphertext and its corresponding plaintext (or to many such pairs). In a [chosen-plaintext attack](https://en.wikipedia.org/wiki/Chosen-plaintext_attack), Eve may choose a plaintext and learn its corresponding ciphertext (perhaps many times); an example is [gardening](https://en.wikipedia.org/wiki/Gardening_(cryptanalysis)), used by the British during WWII. In a [chosen-ciphertext attack](https://en.wikipedia.org/wiki/Chosen-ciphertext_attack), Eve may be able to *choose* ciphertexts and learn their corresponding plaintexts. Finally in a [man-in-the-middle](https://en.wikipedia.org/wiki/Man-in-the-middle_attack)attack Eve gets in between Alice (the sender) and Bob (the recipient), accesses and modifies the traffic and then forwards it to the recipient. Also important, often overwhelmingly so, are mistakes (generally in the design or use of one of the[protocols](https://en.wikipedia.org/wiki/Cryptographic_protocol) involved; see [Cryptanalysis of the Enigma](https://en.wikipedia.org/wiki/Cryptanalysis_of_the_Enigma) for some historical examples of this).

[Poznań](https://en.wikipedia.org/wiki/Pozna%C5%84) monument (*center*) to Polish cryptologists whose breaking of Germany's Enigma machine ciphers, beginning in 1932, altered the course of World War II

Cryptanalysis of symmetric-key ciphers typically involves looking for attacks against the block ciphers or stream ciphers that are more efficient than any attack that could be against a perfect cipher. For example, a simple brute force attack against DES requires one known plaintext and 255 decryptions, trying approximately half of the possible keys, to reach a point at which chances are better than even that the key sought will have been found. But this may not be enough assurance; a [linear cryptanalysis](https://en.wikipedia.org/wiki/Linear_cryptanalysis) attack against DES requires 243 known plaintexts and approximately 243 DES operations.[[42]](https://en.wikipedia.org/wiki/Cryptography#cite_note-junod-42) This is a considerable improvement on brute force attacks.

Public-key algorithms are based on the computational difficulty of various problems. The most famous of these is [integer factorization](https://en.wikipedia.org/wiki/Integer_factorization) (e.g., the RSA algorithm is based on a problem related to integer factoring), but the [discrete logarithm](https://en.wikipedia.org/wiki/Discrete_logarithm) problem is also important. Much public-key cryptanalysis concerns numerical algorithms for solving these computational problems, or some of them, efficiently (i.e., in a practical time). For instance, the best known algorithms for solving the [elliptic curve-based](https://en.wikipedia.org/wiki/Elliptic_curve_cryptography) version of discrete logarithm are much more time-consuming than the best known algorithms for factoring, at least for problems of more or less equivalent size. Thus, other things being equal, to achieve an equivalent strength of attack resistance, factoring-based encryption techniques must use larger keys than elliptic curve techniques. For this reason, public-key cryptosystems based on elliptic curves have become popular since their invention in the mid-1990s.

While pure cryptanalysis uses weaknesses in the algorithms themselves, other attacks on cryptosystems are based on actual use of the algorithms in real devices, and are called [*side-channel attacks*](https://en.wikipedia.org/wiki/Side-channel_attack). If a cryptanalyst has access to, for example, the amount of time the device took to encrypt a number of plaintexts or report an error in a password or PIN character, he may be able to use a [timing attack](https://en.wikipedia.org/wiki/Timing_attack) to break a cipher that is otherwise resistant to analysis. An attacker might also study the pattern and length of messages to derive valuable information; this is known as [traffic analysis](https://en.wikipedia.org/wiki/Traffic_analysis) and can be quite useful to an alert adversary. Poor administration of a cryptosystem, such as permitting too short keys, will make any system vulnerable, regardless of other virtues. And, of course, [social engineering](https://en.wikipedia.org/wiki/Social_engineering_(security)), and other attacks against the personnel who work with cryptosystems or the messages they handle (e.g., [bribery](https://en.wikipedia.org/wiki/Bribery), [extortion](https://en.wikipedia.org/wiki/Extortion), [blackmail](https://en.wikipedia.org/wiki/Blackmail), [espionage](https://en.wikipedia.org/wiki/Espionage), [torture](https://en.wikipedia.org/wiki/Torture), ...) may be the most productive attacks of all.

### CRYPTOGRAPHIC PRIMITIVES

Much of the theoretical work in cryptography concerns [cryptographic *primitives*](https://en.wikipedia.org/wiki/Cryptographic_primitive)—algorithms with basic cryptographic properties—and their relationship to other cryptographic problems. More complicated cryptographic tools are then built from these basic primitives. These primitives provide fundamental properties, which are used to develop more complex tools called *cryptosystems* or *cryptographic protocols*, which guarantee one or more high-level security properties. Note however, that the distinction between cryptographic *primitives* and cryptosystems, is quite arbitrary; for example, the [RSA](https://en.wikipedia.org/wiki/RSA_(algorithm)) algorithm is sometimes considered a cryptosystem, and sometimes a primitive. Typical examples of cryptographic primitives include[pseudorandom functions](https://en.wikipedia.org/wiki/Pseudorandom_function), [one-way functions](https://en.wikipedia.org/wiki/One-way_function), etc.

### CRYPTOSYSTEMS

One or more cryptographic primitives are often used to develop a more complex algorithm, called a cryptographic system, or*cryptosystem*. Cryptosystems (e.g., [El-Gamal encryption](https://en.wikipedia.org/wiki/ElGamal_encryption)) are designed to provide particular functionality (e.g., public key encryption) while guaranteeing certain security properties (e.g., [chosen-plaintext attack (CPA)](https://en.wikipedia.org/wiki/Chosen-plaintext_attack) security in the [random oracle model](https://en.wikipedia.org/wiki/Random_oracle_model)). Cryptosystems use the properties of the underlying cryptographic primitives to support the system's security properties. Of course, as the distinction between primitives and cryptosystems is somewhat arbitrary, a sophisticated cryptosystem can be derived from a combination of several more primitive cryptosystems. In many cases, the cryptosystem's structure involves back and forth communication among two or more parties in space (e.g., between the sender of a secure message and its receiver) or across time (e.g., cryptographically protected [backup](https://en.wikipedia.org/wiki/Backup) data). Such cryptosystems are sometimes called [*cryptographic protocols*](https://en.wikipedia.org/wiki/Cryptographic_protocol).

Some widely known cryptosystems include [RSA encryption](https://en.wikipedia.org/wiki/RSA_(algorithm)), [Schnorr signature](https://en.wikipedia.org/wiki/Schnorr_signature" \o "Schnorr signature), El-Gamal encryption, [PGP](https://en.wikipedia.org/wiki/Pretty_Good_Privacy), etc. More complex cryptosystems include [electronic cash](https://en.wikipedia.org/wiki/Electronic_cash) systems, [signcryption](https://en.wikipedia.org/wiki/Signcryption" \o "Signcryption) systems, etc. Some more 'theoretical' cryptosystems include [interactive proof systems](https://en.wikipedia.org/wiki/Interactive_proof_system), (like [zero-knowledge proofs](https://en.wikipedia.org/wiki/Zero-knowledge_proof)), systems for [secret sharing](https://en.wikipedia.org/wiki/Secret_sharing), etc.

Until recentlymost security properties of most cryptosystems were demonstrated using empirical techniques or using ad hoc reasoning. Recently[there has been considerable effort to develop formal techniques for establishing the security of cryptosystems; this has been generally called [*provable security*](https://en.wikipedia.org/wiki/Provable_security). The general idea of provable security is to give arguments about the computational difficulty needed to compromise some security aspect of the cryptosystem (i.e., to any adversary).

The study of how best to implement and integrate cryptography in software applications is itself a distinct field (see[Cryptographic engineering](https://en.wikipedia.org/wiki/Cryptographic_engineering) and [Security engineering](https://en.wikipedia.org/wiki/Security_engineering)).

## LEGAL ISSUES

### PROHIBITIONS

Cryptography has long been of interest to intelligence gathering and [law enforcement agencies](https://en.wikipedia.org/wiki/Law_enforcement_agency). Secret communications may be criminal or even [treasonous](https://en.wikipedia.org/wiki/Treason)Because of its facilitation of [privacy](https://en.wikipedia.org/wiki/Privacy), and the diminution of privacy attendant on its prohibition, cryptography is also of considerable interest to civil rights supporters. Accordingly, there has been a history of controversial legal issues surrounding cryptography, especially since the advent of inexpensive computers has made widespread access to high quality cryptography possible.

In some countries, even the domestic use of cryptography is, or has been, restricted. Until 1999, [France](https://en.wikipedia.org/wiki/France) significantly restricted the use of cryptography domestically, though it has since relaxed many of these rules. In [China](https://en.wikipedia.org/wiki/People%27s_Republic_of_China) and [Iran](https://en.wikipedia.org/wiki/Islamic_Republic_of_Iran), a license is still required to use cryptography. Many countries have tight restrictions on the use of cryptography. Among the more restrictive are laws in [Belarus](https://en.wikipedia.org/wiki/Belarus), [Kazakhstan](https://en.wikipedia.org/wiki/Kazakhstan), [Mongolia](https://en.wikipedia.org/wiki/Mongolia), [Pakistan](https://en.wikipedia.org/wiki/Pakistan), [Singapore](https://en.wikipedia.org/wiki/Singapore), [Tunisia](https://en.wikipedia.org/wiki/Tunisia), and [Vietnam](https://en.wikipedia.org/wiki/Vietnam).

In the [United States](https://en.wikipedia.org/wiki/United_States), cryptography is legal for domestic use, but there has been much conflict over legal issues related to cryptography. One particularly important issue has been the [export of cryptography](https://en.wikipedia.org/wiki/Export_of_cryptography) and cryptographic software and hardware. Probably because of the importance of cryptanalysis in [World War II](https://en.wikipedia.org/wiki/World_War_II) and an expectation that cryptography would continue to be important for national security, many Western governments have, at some point, strictly regulated export of cryptography. After World War II, it was illegal in the US to sell or distribute encryption technology overseas; in fact, encryption was designated as auxiliary military equipment and put on the [United States Munitions List](https://en.wikipedia.org/wiki/United_States_Munitions_List). Until the development of the [personal computer](https://en.wikipedia.org/wiki/Personal_computer), asymmetric key algorithms (i.e., public key techniques), and the [Internet](https://en.wikipedia.org/wiki/Internet), this was not especially problematic. However, as the Internet grew and computers became more widely available, high-quality encryption techniques became well known around the globe.

### EXPORT CONTROLS

In the 1990s, there were several challenges to US export regulation of cryptography. After the [source code](https://en.wikipedia.org/wiki/Source_code) for [Philip Zimmermann](https://en.wikipedia.org/wiki/Philip_Zimmermann)'s [Pretty Good Privacy](https://en.wikipedia.org/wiki/Pretty_Good_Privacy) (PGP) encryption program found its way onto the Internet in June 1991, a complaint by[RSA Security](https://en.wikipedia.org/wiki/RSA_Security) (then called RSA Data Security, Inc.) resulted in a lengthy criminal investigation of Zimmermann by the US Customs Service and the [FBI](https://en.wikipedia.org/wiki/Federal_Bureau_of_Investigation), though no charges were ever filed. [Daniel J. Bernstein](https://en.wikipedia.org/wiki/Daniel_J._Bernstein), then a graduate student at [UC Berkeley](https://en.wikipedia.org/wiki/UC_Berkeley), brought a lawsuit against the US government challenging some aspects of the restrictions based on [free speech](https://en.wikipedia.org/wiki/1st_Amendment)grounds. The 1995 case [Bernstein v. United States](https://en.wikipedia.org/wiki/Bernstein_v._United_States) ultimately resulted in a 1999 decision that printed source code for cryptographic algorithms and systems was protected as [free speech](https://en.wikipedia.org/wiki/Freedom_of_speech) by the United States Constitution.

In 1996, thirty-nine countries signed the [Wassenaar Arrangement](https://en.wikipedia.org/wiki/Wassenaar_Arrangement" \o "Wassenaar Arrangement), an arms control treaty that deals with the export of arms and "dual-use" technologies such as cryptography. The treaty stipulated that the use of cryptography with short key-lengths (56-bit for symmetric encryption, 512-bit for RSA) would no longer be export-controlled. Cryptography exports from the US became less strictly regulated as a consequence of a major relaxation in 2000; there are no longer very many restrictions on key sizes in US-[exported](https://en.wikipedia.org/wiki/Export_of_cryptography) mass-market software. Since this relaxation in US export restrictions, and because most personal computers connected to the [Internet](https://en.wikipedia.org/wiki/Internet) include US-sourced [web browsers](https://en.wikipedia.org/wiki/Web_browser) such as [Firefox](https://en.wikipedia.org/wiki/Firefox) or [Internet Explorer](https://en.wikipedia.org/wiki/Internet_Explorer), almost every Internet user worldwide has potential access to quality cryptography via their browsers (e.g., via [Transport Layer Security](https://en.wikipedia.org/wiki/Transport_Layer_Security)). The [Mozilla Thunderbird](https://en.wikipedia.org/wiki/Mozilla_Thunderbird) and [Microsoft Outlook](https://en.wikipedia.org/wiki/Microsoft_Outlook) [E-mail client](https://en.wikipedia.org/wiki/E-mail_client) programs similarly can transmit and receive emails via TLS, and can send and receive email encrypted with [S/MIME](https://en.wikipedia.org/wiki/S/MIME). Many Internet users don't realize that their basic application software contains such extensive [cryptosystems](https://en.wikipedia.org/wiki/Cryptosystem). These browsers and email programs are so ubiquitous that even governments whose intent is to regulate civilian use of cryptography generally don't find it practical to do much to control distribution or use of cryptography of this quality, so even when such laws are in force, actual enforcement is often effectively impossible.

### NSA INVOLVEMENT

Another contentious issue connected to cryptography in the United States is the influence of the [National Security Agency](https://en.wikipedia.org/wiki/National_Security_Agency) on cipher development and policy.[[8]](https://en.wikipedia.org/wiki/Cryptography#cite_note-RangerSteve1-8) The NSA was involved with the design of [DES](https://en.wikipedia.org/wiki/Data_Encryption_Standard) during its development at [IBM](https://en.wikipedia.org/wiki/IBM) and its consideration by the [National Bureau of Standards](https://en.wikipedia.org/wiki/National_Bureau_of_Standards) as a possible Federal Standard for cryptography. DES was designed to be resistant to [differential cryptanalysis](https://en.wikipedia.org/wiki/Differential_cryptanalysis), powerful and general cryptanalytic technique known to the NSA and IBM, that became publicly known only when it was rediscovered in the late 1980s. According to [Steven Levy](https://en.wikipedia.org/wiki/Steven_Levy), IBM discovered differential cryptanalysis, but kept the technique secret at the NSA's request. The technique became publicly known only when Biham and Shamir re-discovered and announced it some years later. The entire affair illustrates the difficulty of determining what resources and knowledge an attacker might actually have.

Another instance of the NSA's involvement was the 1993 [Clipper chip](https://en.wikipedia.org/wiki/Clipper_chip) affair, an encryption microchip intended to be part of the [Capstone](https://en.wikipedia.org/wiki/Capstone_(cryptography)) cryptography-control initiative. Clipper was widely criticized by cryptographers for two reasons. The cipher algorithm (called [Skipjack](https://en.wikipedia.org/wiki/Skipjack_(cipher))) was then classified (declassified in 1998, long after the Clipper initiative lapsed). The classified cipher caused concerns that the NSA had deliberately made the cipher weak in order to assist its intelligence efforts. The whole initiative was also criticized based on its violation of [Kerckhoffs's Principle](https://en.wikipedia.org/wiki/Kerckhoffs%27s_Principle" \o "Kerckhoffs's Principle), as the scheme included a special [escrow key](https://en.wikipedia.org/wiki/Key_escrow) held by the government for use by law enforcement, for example in wiretaps.

### DIGITAL RIGHTS MANAGEMENT

Cryptography is central to digital rights management (DRM), a group of techniques for technologically controlling use of[copyrighted](https://en.wikipedia.org/wiki/Copyright) material, being widely implemented and deployed at the behest of some copyright holders. In 1998, [U.S. President](https://en.wikipedia.org/wiki/U.S._President) [Bill Clinton](https://en.wikipedia.org/wiki/Bill_Clinton) signed the [Digital Millennium Copyright Act](https://en.wikipedia.org/wiki/Digital_Millennium_Copyright_Act) (DMCA), which criminalized all production, dissemination, and use of certain cryptanalytic techniques and technology (now known or later discovered); specifically, those that could be used to circumvent DRM technological schemes. This had a noticeable impact on the cryptography research community since an argument can be made that *any* cryptanalytic research violated, or might violate, the DMCA. Similar statutes have since been enacted in several countries and regions, including the implementation in the [EU Copyright Directive](https://en.wikipedia.org/wiki/Directive_on_the_harmonisation_of_certain_aspects_of_copyright_and_related_rights_in_the_information_society). Similar restrictions are called for by treaties signed by [World Intellectual Property Organization](https://en.wikipedia.org/wiki/World_Intellectual_Property_Organization) member-states.

The [United States Department of Justice](https://en.wikipedia.org/wiki/United_States_Department_of_Justice) and [FBI](https://en.wikipedia.org/wiki/Federal_Bureau_of_Investigation) have not enforced the DMCA as rigorously as had been feared by some, but the law, nonetheless, remains a controversial one. [Niels Ferguson](https://en.wikipedia.org/wiki/Niels_Ferguson" \o "Niels Ferguson), a well-respected cryptography researcher, has publicly stated that he will not release some of his research into an [Intel](https://en.wikipedia.org/wiki/Intel_Corporation) security design for fear of prosecution under the DMCA. Cryptanalyst [Bruce Schneier](https://en.wikipedia.org/wiki/Bruce_Schneier) has argued that the DMCA encourages [vendor lock-in](https://en.wikipedia.org/wiki/Vendor_lock-in), while inhibiting actual measures toward cyber-security. Both [Alan Cox](https://en.wikipedia.org/wiki/Alan_Cox) (longtime [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel) developer) and [Edward Felten](https://en.wikipedia.org/wiki/Edward_Felten) (and some of his students at Princeton) have encountered problems related to the Act. [Dmitry Sklyarov](https://en.wikipedia.org/wiki/Dmitry_Sklyarov) was arrested during a visit to the US from Russia, and jailed for five months pending trial for alleged violations of the DMCA arising from work he had done in Russia, where the work was legal. In 2007, the cryptographic keys responsible for [Blu-ray](https://en.wikipedia.org/wiki/Blu-ray) and [HD DVD](https://en.wikipedia.org/wiki/HD_DVD) content scrambling were [discovered and released onto the Internet](https://en.wikipedia.org/wiki/AACS_encryption_key_controversy). In both cases, the [MPAA](https://en.wikipedia.org/wiki/MPAA) sent out numerous DMCA takedown notices, and there was a massive Internet backlashtriggered by the perceived impact of such notices on [fair use](https://en.wikipedia.org/wiki/Fair_use) and [free speech](https://en.wikipedia.org/wiki/Free_speech).

### FORCED DISCLOSURE OF ENCRYPTION KEYS

In the United Kingdom, the [Regulation of Investigatory Powers Act](https://en.wikipedia.org/wiki/Regulation_of_Investigatory_Powers_Act_2000) gives UK police the powers to force suspects to decrypt files or hand over passwords that protect encryption keys. Failure to comply is an offense in its own right, punishable on conviction by a two-year jail sentence or up to five years in cases involving national security. Successful prosecutions have occurred under the Act; the first, in 2009, resulted in a term of 13 months' imprisonment. Similar forced disclosure laws in Australia, Finland, France, and India compel individual suspects under investigation to hand over encryption keys or passwords during a criminal investigation.

In the United States, the federal criminal case of [United States v. Fricosu](https://en.wikipedia.org/wiki/United_States_v._Fricosu) addressed whether a search warrant can compel a person to reveal an [encryption](https://en.wikipedia.org/wiki/Encryption) [passphrase](https://en.wikipedia.org/wiki/Passphrase) or password. The [Electronic Frontier Foundation](https://en.wikipedia.org/wiki/Electronic_Frontier_Foundation) (EFF) argued that this is a violation of the protection from self-incrimination given by the [Fifth Amendment](https://en.wikipedia.org/wiki/Fifth_Amendment_to_the_United_States_Constitution). In 2012, the court ruled that under the [All Writs Act](https://en.wikipedia.org/wiki/All_Writs_Act), the defendant was required to produce an unencrypted hard drive for the court.

In many jurisdictions, the legal status of forced disclosure remains unclear.

The 2016 [FBI–Apple encryption dispute](https://en.wikipedia.org/wiki/FBI%E2%80%93Apple_encryption_dispute) concerns the ability of courts in the United States to compel manufacturers' assistance in unlocking cell phones whose contents are cryptographically protected.

As a potential counter-measure to forced disclosure some cryptographic software supports [plausible deniability](https://en.wikipedia.org/wiki/Plausible_deniability), where the encrypted data is indistinguishable from unused random data (for example such as that of a drive which has been securely wiped).

CLAUDE SHANNON

[Claude E. Shannon](https://en.wikipedia.org/wiki/Claude_Shannon) is considered by many to be the father of mathematical cryptography. Shannon worked for several years at Bell Labs, and during his time there, he produced an article entitled “A mathematical theory of cryptography”. This article was written in 1945 and eventually was published in the Bell System Technical Journal in 1949. It is commonly accepted that this paper was the starting point for development of modern cryptography. Shannon was inspired during the war to address “[t]he problems of cryptography [because] secrecy systems furnish an interesting application of communication theory”. Shannon identified the two main goals of cryptography: secrecy and authenticity. His focus was on exploring secrecy and thirty-five years later, G.J. Simmons would address the issue of authenticity. Shannon wrote a further article entitled “A mathematical theory of communication” which highlights one of the most significant aspects of his work: cryptography’s transition from art to science.

In his works, Shannon described the two basic types of systems for secrecy. The first are those designed with the intent to protect against hackers and attackers who have infinite resources with which to decode a message (theoretical secrecy, now unconditional security), and the second are those designed to protect against hackers and attacks with finite resources with which to decode a message (practical secrecy, now computational security). Most of Shannon’s work focused around theoretical secrecy; here, Shannon introduced a definition for the “unbreakability” of a cipher. If a cipher was determined “unbreakable”, it was considered to have “perfect secrecy”. In proving “perfect secrecy”, Shannon determined that this could only be obtained with a secret key whose length given in binary digits was greater than or equal to the number of bits contained in the information being encrypted. Furthermore, Shannon developed the “unicity distance”, defined as the “amount of plaintext that… determines the secret key.”

Shannon’s work influenced further cryptography research in the 1970s, as the public-key cryptography developers, M. E. Hellman and W. Diffie cited Shannon’s research as a major influence. His work also impacted modern designs of secret-key ciphers. At the end of Shannon’s work with cryptography, progress slowed until Hellman and Diffie introduced their paper involving “public-key cryptography”.

### AN ENCRYPTION STANDARD

The mid-1970s saw two major public (i.e., non-secret) advances. First was the publication of the draft [Data Encryption Standard](https://en.wikipedia.org/wiki/Data_Encryption_Standard) in the U.S. *Federal Register* on 17 March 1975. The proposed DES cipher was submitted by a research group at[IBM](https://en.wikipedia.org/wiki/International_Business_Machines), at the invitation of the National Bureau of Standards (now [NIST](https://en.wikipedia.org/wiki/NIST)), in an effort to develop secure electronic communication facilities for businesses such as banks and other large financial organizations. After advice and modification by the [NSA](https://en.wikipedia.org/wiki/NSA), acting behind the scenes, it was adopted and published as a [Federal Information Processing Standard](https://en.wikipedia.org/wiki/Federal_Information_Processing_Standard)Publication in 1977 (currently at [FIPS 46-3](http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf)). DES was the first publicly accessible cipher to be 'blessed' by a national agency such as the NSA. The release of its specification by NBS stimulated an explosion of public and academic interest in cryptography.

The aging DES was officially replaced by the [Advanced Encryption Standard](https://en.wikipedia.org/wiki/Advanced_Encryption_Standard) (AES) in 2001 when NIST announced FIPS 197. After an open competition, NIST selected [Rijndael](https://en.wikipedia.org/wiki/Rijndael" \o "Rijndael), submitted by two Belgian cryptographers, to be the AES. DES, and more secure variants of it (such as [Triple DES](https://en.wikipedia.org/wiki/Triple_DES)), are still used today, having been incorporated into many national and organizational standards. However, its 56-bit key-size has been shown to be insufficient to guard against [brute force attacks](https://en.wikipedia.org/wiki/Brute_force_attack)(one such attack, undertaken by the cyber civil-rights group [Electronic Frontier Foundation](https://en.wikipedia.org/wiki/Electronic_Frontier_Foundation) in 1997, succeeded in 56 hours.) As a result, use of straight DES encryption is now without doubt insecure for use in new cryptosystem designs, and messages protected by older cryptosystems using DES, and indeed all messages sent since 1976 using DES, are also at risk. Regardless of DES' inherent quality, the DES key size (56-bits) was thought to be too small by some even in 1976, perhaps most publicly by [Whitfield Diffie](https://en.wikipedia.org/wiki/Whitfield_Diffie). There was suspicion that government organizations even then had sufficient computing power to break DES messages; clearly others have achieved this capability.

### PUBLIC KEY

The second development, in 1976, was perhaps even more important, for it fundamentally changed the way cryptosystems might work. This was the publication of the paper [New Directions in Cryptography](http://www-ee.stanford.edu/~hellman/publications/24.pdf) by [Whitfield Diffie](https://en.wikipedia.org/wiki/Whitfield_Diffie) and [Martin Hellman](https://en.wikipedia.org/wiki/Martin_Hellman). It introduced a radically new method of distributing cryptographic keys, which went far toward solving one of the fundamental problems of cryptography, key distribution, and has become known as [Diffie-Hellman key exchange](https://en.wikipedia.org/wiki/Diffie-Hellman_key_exchange" \o "Diffie-Hellman key exchange). The article also stimulated the almost immediate public development of a new class of enciphering algorithms, the [asymmetric key algorithms](https://en.wikipedia.org/wiki/Asymmetric_key_algorithm).

Prior to that time, all useful modern encryption algorithms had been [symmetric key algorithms](https://en.wikipedia.org/wiki/Symmetric_key_algorithm), in which the same[cryptographic key](https://en.wikipedia.org/wiki/Cryptographic_key) is used with the underlying algorithm by both the sender and the recipient, who must both keep it secret. All of the electromechanical machines used in World War II were of this logical class, as were the [Caesar](https://en.wikipedia.org/wiki/Caesar_cipher) and [Atbash](https://en.wikipedia.org/wiki/Atbash" \o "Atbash) ciphers and essentially all cipher systems throughout history. The 'key' for a code is, of course, the codebook, which must likewise be distributed and kept secret, and so shares most of the same problems in practice.

Of necessity, the key in every such system had to be exchanged between the communicating parties in some secure way prior to any use of the system (the term usually used is 'via a [secure channel](https://en.wikipedia.org/wiki/Secure_channel)') such as a trustworthy courier with a briefcase handcuffed to a wrist, or face-to-face contact, or a loyal carrier pigeon. This requirement is never trivial and very rapidly becomes unmanageable as the number of participants increases, or when secure channels aren't available for key exchange, or when, as is sensible cryptographic practice, keys are frequently changed. In particular, if messages are meant to be secure from other users, a separate key is required for each possible pair of users. A system of this kind is known as a secret key, or [symmetric key](https://en.wikipedia.org/wiki/Symmetric_key) cryptosystem. D-H key exchange (and succeeding improvements and variants) made operation of these systems much easier, and more secure, than had ever been possible before in all of history.

In contrast, [asymmetric key](https://en.wikipedia.org/wiki/Asymmetric_key_algorithm) encryption uses a pair of mathematically related keys, each of which decrypts the encryption performed using the other. Some, but not all, of these algorithms have the additional property that one of the paired keys cannot be deduced from the other by any known method other than trial and error. An algorithm of this kind is known as a public key or [asymmetric key](https://en.wikipedia.org/wiki/Asymmetric_key_cryptography) system. Using such an algorithm, only one key pair is needed per user. By designating one key of the pair as private (always secret), and the other as public (often widely available), no secure channel is needed for key exchange. So long as the private key stays secret, the public key can be widely known for a very long time without compromising security, making it safe to reuse the same key pair indefinitely.

For two users of an asymmetric key algorithm to communicate securely over an insecure channel, each user will need to know their own public and private keys as well as the other user's public key. Take this basic scenario: [Alice and Bob](https://en.wikipedia.org/wiki/Alice_and_Bob) each have a pair of keys they've been using for years with many other users. At the start of their message, they exchange public keys, unencrypted over an insecure line. Alice then encrypts a message using her private key, and then re-encrypts that result using Bob's public key. The double-encrypted message is then sent as digital data over a wire from Alice to Bob. Bob receives the bit stream and decrypts it using his own private key, and then decrypts that bit stream using Alice's public key. If the final result is recognizable as a message, Bob can be confident that the message actually came from someone who knows Alice's private key (presumably actually her if she's been careful with her private key), and that anyone eavesdropping on the channel will need Bob's private key in order to understand the message.

Asymmetric algorithms rely for their effectiveness on a class of problems in mathematics called one-way functions, which require relatively little computational power to execute, but vast amounts of power to reverse, if reversal is possible at all. A classic example of a one-way function is multiplication of very large prime numbers. It's fairly quick to multiply two large primes, but very difficult to find the factors of the product of two large primes. Because of the mathematics of one-way functions, most possible keys are bad choices as cryptographic keys; only a small fraction of the possible keys of a given length are suitable, and so asymmetric algorithms require very long keys to reach the same [level of security](https://en.wikipedia.org/wiki/Level_of_security) provided by relatively shorter symmetric keys. The need to both generate the key pairs, and perform the encryption/decryption operations make asymmetric algorithms computationally expensive, compared to most symmetric algorithms. Since symmetric algorithms can often use any sequence of (random, or at least unpredictable) bits as a key, a disposable *session key* can be quickly generated for short-term use. Consequently, it is common practice to use a long asymmetric key to exchange a disposable, much shorter (but just as strong) symmetric key. The slower asymmetric algorithm securely sends a symmetric session key, and the faster symmetric algorithm takes over for the remainder of the message.

Asymmetric key cryptography, Diffie-Hellman key exchange, and the best known of the public key / private key algorithms (i.e., what is usually called the RSA algorithm), all seem to have been independently developed at a UK intelligence agency before the public announcement by Diffie and Hellman in 1976. GCHQ has released documents claiming they had developed public key cryptography before the publication of Diffie and Hellman's paper.[[*citation needed*](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed)] Various classified papers were written at GCHQ during the 1960s and 1970s which eventually led to schemes essentially identical to RSA encryption and to Diffie-Hellman key exchange in 1973 and 1974. Some of these have now been published, and the inventors (James H. Ellis, Clifford Cocks, and Malcolm Williamson) have made public (some of) their work.

### HASHING

[Hashing](https://en.wikipedia.org/wiki/Hash_function) is a common technique used in cryptography to encode information quickly using typical algorithms. Generally, an[algorithm](https://en.wikipedia.org/wiki/Algorithm) is applied to a string of text, and the resulting string becomes the “hash value”. This creates a “digital fingerprint” of the message, as the specific hash value is used to identify a specific message. The output from the algorithm is also referred to as a “message digest” or a “check sum”. Hashing is good for determining if information has been changed in transmission. If the hash value is different upon reception than upon sending, there is evidence the message has been altered. Once the algorithm has been applied to the data to be hashed, the hash function produces a fixed-length output. Essentially, anything passed through the hash function should resolve to the same length output as anything else passed through the same hash function. It is important to note that hashing is not the same as encrypting. Hashing is a one-way operation that is used to transform data into the compressed message digest. Additionally, the integrity of the message can be measured with hashing. Conversely, encryption is a two-way operation that is used to transform plaintext into cipher-text and then vice versa. In encryption, the confidentiality of a message is guaranteed.

Hash functions can be used to verify digital signatures, so that when signing documents via the Internet, the signature is applied to one particular individual. Much like a hand-written signature, these signatures are verified by assigning their exact hash code to a person. Furthermore, hashing is applied to passwords for computer systems. Hashing for passwords began with the [UNIX](https://en.wikipedia.org/wiki/Unix) operating system. A user on the system would first create a password. That password would be hashed, using an algorithm or key, and then stored in a password file. This is still prominent today, as web applications that require passwords will often hash user’s passwords and store them in a database.

### CRYPTOGRAPHY POLITICS

The public developments of the 1970s broke the near monopoly on high quality cryptography held by government organizations (see S Levy's crypto for a journalistic account of some of the policy controversy of the time in the US). For the first time ever, those outside government organizations had access to cryptography not readily breakable by anyone (including governments). Considerable controversy, and conflict, both public and private, began more or less immediately, sometimes called the [crypto wars](https://en.wikipedia.org/wiki/Crypto_wars). They have not yet subsided. In many countries, for example, [export of cryptography](https://en.wikipedia.org/wiki/Export_of_cryptography) is subject to restrictions. Until 1996 export from the U.S. of cryptography using keys longer than 40 bits (too small to be very secure against a knowledgeable attacker) was sharply limited. As recently as 2004, former [FBI](https://en.wikipedia.org/wiki/FBI) Director [Louis Freeh](https://en.wikipedia.org/wiki/Louis_Freeh), testifying before the [9/11 Commission](https://en.wikipedia.org/wiki/9/11_Commission), called for new laws against public use of encryption.

One of the most significant people favoring strong encryption for public use was [Phil Zimmermann](https://en.wikipedia.org/wiki/Phil_Zimmermann). He wrote and then in 1991 released [PGP](https://en.wikipedia.org/wiki/Pretty_Good_Privacy) (Pretty Good Privacy), a very high quality [crypto system](https://en.wikipedia.org/wiki/Crypto_system). He distributed a freeware version of PGP when he felt threatened by legislation then under consideration by the US Government that would require backdoors to be included in all cryptographic products developed within the US. His system was released worldwide shortly after he released it in the US, and that began a long criminal investigation of him by the US Government Justice Department for the alleged violation of export restrictions. The Justice Department eventually dropped its case against Zimmermann, and the freeware distribution of PGP has continued around the world. PGP even eventually became an open [Internet](https://en.wikipedia.org/wiki/IETF) standard ([RFC 2440](https://tools.ietf.org/html/rfc2440)or [OpenPGP](https://en.wikipedia.org/wiki/OpenPGP" \o "OpenPGP)).

### MODERN CRYPTANALYSIS

While modern ciphers like [AES](https://en.wikipedia.org/wiki/Advanced_Encryption_Standard) and the higher quality asymmetric ciphers are widely considered unbreakable, poor designs and implementations are still sometimes adopted and there have been important cryptanalytic breaks of deployed crypto systems in recent years. Notable examples of broken crypto designs include the first [Wi-Fi](https://en.wikipedia.org/wiki/Wi-Fi) encryption scheme [WEP](https://en.wikipedia.org/wiki/Wired_Equivalent_Privacy), the[Content Scrambling System](https://en.wikipedia.org/wiki/Content_Scrambling_System) used for encrypting and controlling DVD use, the [A5/1](https://en.wikipedia.org/wiki/A5/1) and [A5/2](https://en.wikipedia.org/wiki/A5/2) ciphers used in [GSM](https://en.wikipedia.org/wiki/GSM) cell phones, and the [CRYPTO1](https://en.wikipedia.org/wiki/CRYPTO1) cipher used in the widely deployed [MIFARE](https://en.wikipedia.org/wiki/MIFARE) Classic [smart cards](https://en.wikipedia.org/wiki/Smart_card) from [NXP Semiconductors](https://en.wikipedia.org/wiki/NXP_Semiconductors), a spun off division of [Philips Electronics](https://en.wikipedia.org/wiki/Philips_Electronics). All of these are symmetric ciphers. Thus far, not one of the mathematical ideas underlying public key cryptography has been proven to be 'unbreakable', and so some future mathematical analysis advance might render systems relying on them insecure. While few informed observers foresee such a breakthrough, the key size recommended for security as best practice keeps increasing as increased computing power required for breaking codes becomes cheaper and more available.

**TYPES OF CRYPTOGRAPHY :**

There are mainly two types of Cryptography :

1)Symmetric Key Cryptography

2)Asymmetric Key Cryptography

**Symmetric Key Cryptography** (Secret Key Cryptography) :

In Symmetric Key Cryptography , Same Key is used by both parties

**Advantages :** Simpler and Faster

**Disadvantages :** Less Secured

**Asymmetric Key Cryptography** (Public Key Cryptography)

In Asymmetric Key Cryptography , 2 different keys are used

Here , Users get the Key from an Certificate Authority

**Advantages :** More Secured and Authentication

**Disadvantages :** Relatively Complex

**CRYPTOGRAPHIC ALGORITHMS :**

1. **DES (Data Encryption Standard ) :**

DES is a previously pre-dominant algorithm used for Encryption/Decryption of electronic data.It is a private key cryptographic algorithm , meaning both the sender and receiver must know and use the same private key .It uses a 56-bit encryption key which can give around 2^56 (256) combinations to encrypt the plain text . DES is restricted with a block size of 64 bits DES is the archetypal block cipher—an algorithm that takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into another ciphertext bit string of the same length. In the case of DES, the block size is 64 bits. DES also uses a key to customize the transformation, so that decryption can supposedly only be performed by those who know the particular key used to encrypt. The key ostensibly consists of 64 bits; however, only 56 of these are actually used by the algorithm. Eight bits are used solely for checking parity, and are thereafter discarded. Hence the effective key length is 56 bits.

The key is nominally stored or transmitted as 8 bytes, each with odd parity. According to ANSI X3.92-1981 (Now, known as ANSI INCITS 92-1981), section 3.5:

One bit in each 8-bit byte of the KEY may be utilized for error detection in key generation, distribution, and storage. Bits 8, 16, 64 are for use in ensuring that each byte is of odd parity.

Like other block ciphers, DES by itself is not a secure means of encryption but must instead be used in a mode of operation. FIPS-81 specifies several modes for use with DES. Further comments on the usage of DES are contained in FIPS-74

Decryption uses the same structure as encryption but with the keys used in reverse order. (This has the advantage that the same hardware or software can be used in both directions.)

1. **AES(Advanced Encryption Standard) :**

Like DES, AES is a symmetric block cipher. This means that it uses the same key for both encryption and decryption. However, AES is quite different from DES in a number of ways. The algorithm Rijndael allows for a variety of block and key sizes and not just the 64 and 56 bits of DES’ block and key size. The block and key can in fact be chosen independently from 128,160,192,224,256 bits and need not be the same. However, the AES standard states that the algorithm can only accept a block size of 128 bits and a choice of three keys - 128,192,256 bits. Depending on which version is used, the name of the standard is modiﬁed to AES-128, AES-192 or AES256 respectively. As well as these differences AES differs from DES in that it is not a feistel structure. Recall that in a feistel structure, half of the data block is used to modify the other half of the data block and then the halves are swapped. In this case the entire data block is processed in parallel during each round using substitutions and permutations.

A number of AES parameters depend on the key length. For example, if the key size used is 128 then the number of rounds is 10 whereas it is 12 and 14 for 192 and 256 bits respectively. At present the most common key size likely to be used is the 128 bit key. This description of the AES algorithm therefore describes this particular implementation.

Rijndael was designed to have the following characteristics:

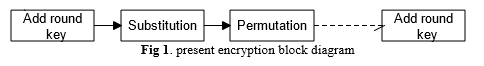
• Resistance against all known attacks.

• Speed and code compactness on a wide range of platforms.

• Design Simplicity..

**Present algorithm**

The present algorithm uses 64 bit plain text and 80 or 128 bit key length. I have taken key size of length 80 bits. Present has SPN structure which is mainly substitution and permutation network. It consists of 31 rounds and final round. One regular round involves key mixing, substitution and permutation layer. We can divide the algorithm into two divisions one is iteration operation and the other one is key updating. Iteration consists of three operations Add round key, S box substitution and permutation layer. Updated key will be used in the add round key operation each time in the encryption. Key updating is used to generate round key each time for add round key operation. Transformations operate on intermediate results are named as state. There are 31 iteration rounds and in the final round intermediate result is XORed with round key to get cipher text. Add round key is also called as key mixing step.



The steps are as follows

1. Generate the round keys ( Key updating)

2. Round key will be XOR ed with the state

3. For 31 rounds substitute s box on the state

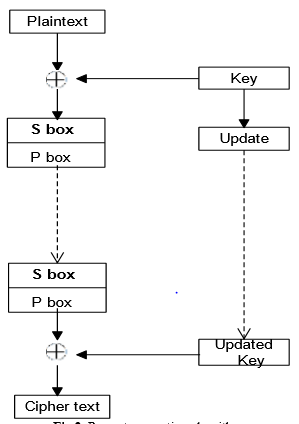
4. For 31 rounds apply permutation layer on the state

5. For 32nd round only Add round key operates

**Key updating**: It involves 61 bit left rotation with substitution box and 5 bit round counter. Leftmost 4 bits are passed through s box. Least significant rightmost values of round counter are XOR ed with bits of K (K19 K18 K17 K16 K15). Present 80 uses single s box and present 128 uses two s boxes.

**S box substitution:**  Substitution layer consists of 16 s boxes with 4 bit inputs and 4 bit outputs. 8 bit s boxes require more area than 4 bit s box. Even though 4 bit s boxes are weaker than 8 bit s boxes we can achieve more security in 4 bit s box. Each S box has 4 bit input.

**Permutation layer or P layer:**  It is simple and regular transformation of bit transposition. Every bit is moved to corresponding bit position as in the permutation layer table. The bit i is moved to the position P(i). For example bit 0 is moved to 0th position and every 4th bits position increases by one. As bits on left increases bit position will be sum of the previous bit and 16 and it starts for every 4 8 12 16 and so on



**KEY-EXCHANGE MANAGEMENT:**

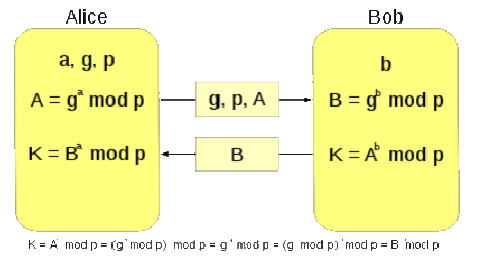
**Diffie-Hellman Key Exchange**:

The Diffie-Hellman Key Exchange is a cryptographic protocol that allows two parties with no prior knowledge of each other to establish a shared secret key, which typically cipher. The Diffie-Hellman Key Exchange was first published by Whitfield Diffie and Martin Hellman in 1976. The GCHQ, the British signals intelligence, announced that this scheme had been invented by Malcolm Williamson years before Diffie and Hellman's publication, but was kept classified.

The Diffie-Hellman Key Exchange relies on exponential functions computing much faster than discrete logarithms. When used properly, the Diffie key without actually transmitting it. The strength of this algorithm depends on the time it takes to compute a discrete logarithm of the public keys transmitted (Diffie, Hellman 1976).

Figure shows the steps for establishing a key through the Diffie wanting to establish a key with Bob, sending the public keys, or numbers, each party can lock that is engaged and disengaged by different keys.

Hellman Key Exchange relies on exponential functions computing much faster than discrete logarithms. When used properly, the Diffie-Hellman Key Exchange protocol gives two parties key without actually transmitting it. The strength of this algorithm depends on the time it takes to compute a discrete logarithm of the public keys transmitted (Diffie, Hellman 1976).



**Diffie-Hellman Key Exchange protocol.**

shows the steps for establishing a key through the Diffie-Hellman Key Exchange. Alice wanting to establish a key with Bob, first sets up the variables "a", "g", and "p." Bob decides "b". sending the public keys, or numbers, each party can compute "K." Notice that "K" is never sent through the medium. Also notice that "K" was not previously determined, rather it was result to both Alice and Bob's computations. This allows each party access to the same key without ever having to see each other. A disadvantage of the Diffie-Hellman key exchange is that it does not contain the function of encryption. A predetermined message cannot be inserted into the algorithm. The transmitted number is simply the result of computation, of which is purposely hard to decompose. In order for "K" to be discovered by someone besides Alice and Bob, a logarithm of "A" or "B" must be computed. When extremely large numbers for "a", "b", and "p" are chosen, it could take billions of years to compute the logarithm of "A" or "B."

**Cryptanalysis**

The objective of cryptanalysis is to obtain as much information as possibleabout hidden aspects of the cipher, i.e. the original data or the secret key.In this section, we classify cryptanalytic methods based on the type ofinformation available to the adversary. Then we introduce complexity parameters to measure the resources required to mount an attack.

**Cryptanalysis Scenarios**

The attacker is assumed to have access to different types of information.In each attack, it should be precisely clarified what kind and how muchof data is required to perform the attack. Data requirement can be aninsurmountable bottleneck for an attack. We can outline a hierarchy ofpossible scenarios based on the type of data available to the attacker.

**Ciphertext-only cryptanalysis**: The attacker has access only to a certain number of ciphertexts.

**Known-plaintext cryptanalysis**: The attacker has access to a limitednumber of plaintexts and their corresponding ciphertexts.

**Chosen-plaintext (ciphertext) cryptanalysis**: The attacker can se-

lect a number of plaintexts (ciphertexts) and query the corresponding ci-

phertexts (plaintexts).

**Adaptively chosen plaintext (ciphertext) cryptanalysis**: The at-

tacker can choose a number of plaintexts (ciphertexts) and ask the cor-

responding ciphertexts (plaintexts) while he has access to the previous

plaintext-ciphertext pairs at each step before choosing the next query.

**Related key**: The attacker has access to a quantity of plaintext-ciphertext

pairs under unknown keys that have a known relationship.

**LIGHT WEIGHT CRYPTOGRAPHY**

Lightweight Cryptography is the collection of cryptographic primitives, techniques and ciphers that can be implemented in highly resource-constrained mobile devices. Light Weight Cryptographic Algorithms are ultra-light weight block ciphers like :

1. Present Encryption Algorithm.
2. Tiny Encryption Algorithm.
3. Scalable Encryption Algorithm.
4. Hight Encryption Algorithm.

Lightweight cryptography is a trade-off between lightweightness and security.In order to reach high levels of security using only a small computing power, many cryptographers have addressed these issues by suggesting lightweight streamciphers, blockciphers, hashfunction and recently one-pass authenticated encryption.These algorithms provide low-resource hardware implementation , which is proper to ubiquitous computing device such as a sensor in USN or Radio Frequency Identification (RFID) tag.In the lightweight context, designer has to analyze the computational complexity of the algorithm, with respect to the demands on the hardware and other limitations of the device.There are both a direction and constraining challenge in these limitations that guide the development of cryptography.Roughly, the weight of “lightweight “primitive is the amount of resources necessary both time and space for it to run. This weight can be measured in two distinct contexts: in software and in hardware.Lightweightness in software does not imply lightweightness in hardware and vice-versa. Finally, a measure which is relevant in both contexts is the power consumption.

CIPHER:

In cryptography, a **cipher** (or **cypher**) is an algorithm for performing encryption or decryption—a series of well-defined steps that can be followed as a procedure

**Ceaser cipher:** this is encryption technique which uses fixed length of alphabets to encrypt.  
**Juilus cipher:** this is encryption technique which uses shift of three alphabets.

**block ciphers**:which encrypt block of data of fixed size, and

**stream cipher:** which encrypt continuous streams of data  
**Feistel ciphers:** are a special class of iterated block ciphers where the cipher text is calculated from the plain text by repeated application of the same transformation or round function.In a Feistel cipher, the text being encrypted is split into two halves. The round function, F, is applied to one half using a sub key and the output of F is (exclusive-or-ed (XORed)) with the other half. The two halves are then swapped. Each round follows the same pattern except for the last round where there is often no swap

**TINY ENCRYPTION ALGORITHM (TEA)**

INTRODUCTION

The Tiny Encryption Algorithm (TEA) is one of the fastest and most efficient cryptography algorithms in existence. The Tiny Encryption Algorithm (TEA) is a symmetric (private) key encryption algorithm created by David Wheeler and Roger Needham of Cambridge University and published in 1994. TEA is a symmetric key algorithm. TEA is designed memory footprint and maximize speed. It is a Feistel type Cipher. Tiny Encryption Algorithm has 32 rounds of simple processes which are shifts, additions and XOR’s. Tiny Encryption Algorithm has 128-bit key length and 64-bit block size. It performs operations on 32 bit words. Tea uses 128-bit key and a magic constant is also utilized which is defines as 2^32/(the golden ratio). This quantity looks like 2654435769 when expressed as an integer. Multiples of this integer are used during each round , and its inclusion in the algorithm was to prevent attacks that try to take advantage of symmetry between rounds. The Tiny Encryption Algorithm (TEA) is a block cipher notable for its simplicity of description and implementation, typically a few lines of code. TEA has fast execution time, and needs minimal storage space. TEA is very secure. There accept been no known successful crypt analyses of TEA. It's conceived to be as secure as the IDEA algorithm, designed by Massey and Xuejia Lai. It uses the same mixed algebraic groups technique as IDEA, but it's very much simpler, hence faster.

**Encryption Methodology**

At the encryption site, TEA takes 64 (block size) data bits time using a 128-bit key with 32 rounds. TEA is an iteration cipher, where each round i has plain text inputs P0 [i-1] and P1 [i-1], which is derived from the previous round. The subkey K[*i*] is derived from the 128 bit overall K and it uses a constant delta (∂), is the derivative of the golden number ratio to ensure that the sub keys are distinct.

The inputs to the encryption algorithm are a plaintext block and a key K. The Plaintext is P=(Left[0],Right[0]) and the cipher text is C=(Left[64],Right[64]). The plain text block is split into two halves, Left[0] and Right [0].Each half is used to encrypt the other half over 64 rounds of processing and then combine to produce the cipher text block.

Each round i has inputs Left[i-1] and Right[i-1],derived from the previous round, as well as a sub key K[i] derived from the 128 bit overall K. The sub key k[i] derived from K and from each other. The constant delta (∂)=(5-1)\*231=9E3779B9h, is derived from the golden number ratio to ensure that the sub keys are distinct and its precise value has no cryptographic significance. The round addition function differs slightly from a classical Fiestel cipher structure in that integer addition modulo 232 is used instead of exclusive-or as the combining operator.

The outputs of each iteration are given by

P0[*i*] = P0[*i-1*] Σ F(P1[*i-1*], K[0, 1], DELTA[*i*])

P1[*i*] = P1[*i-1*] Σ F(P0[*i-1*], K[2, 3], DELTA[*i*])

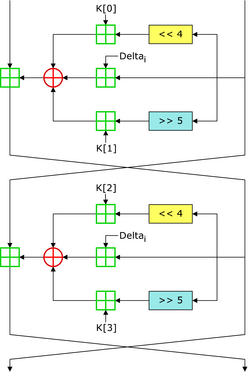
The round function F is defined by

F([P,K[j,k],DELTA[i]) = ((P<<4) Σ K[j]) XOR (P ΣDELTA[i]) XOR ((P>>5) ΣK[k] )

* The single TEA round function performs the simple mixed orthogonal algebraic functions such as Right/Left shifts, Integer addition and exclusive – or operations. The steps carried out in round function:
  + The one half P1 [i-1] of the block cipher is Left shifted by 4 times and Right shifted 5 times.
  + The left shifted block is added with the sub key K0 and right shifted block is added with the sub key K1.
  + It is also added to the constant delta value DELTA[i] which is the multiples of delta, where i represents the number of iterations.
  + The results are then Ex–ORed and added with the other half of the block cipher P0[i-1] which produces one half of the block cipher MO for the next iteration.
  + Similar operations are performed for the next half round function with the above result.

Finally, the Ex–OR ed result is added with the first half of the block cipher P1[i-1] to produce the next half block M1 for the further rounds.

**Decryption Methodology:**



Similar operations are performed for decryption process which is described in figure 2.

In this case, the constant delta value DELTA(i-1) is “C6EF3720”, where „i‟ represents the number of iterations. In each iteration, the delta value “9E3779B9” is subtracted with the constant delta value. To decrypt the encrypted data the reverse operation of the encryption process can be done since TEA uses Fiestel structure.

These two Feistal rounds make up one cycle of TEA. The cipher starts with a 64 bit data block that is split up into two 32 bit blocks which we will call L and R. L is the left side of the block (represented by the arrow in the top left of the diagram), and R is the right side (top right of the diagram). These blocks are swapped per round; this swap can be seen where the two lines intersect in the middle of the diagram.

TEA has a 128 bit key that is split up into four 32 bit subkeys, which can be seen as K[03]

in the diagram. Delta is defined as a constant, 2^32/(golden ratio), which is 2654435769 as an integer. Multiples of delta are used in each round (mod 2^32). In the first feistel round R is used as an input to several operations. All addition operations are (mod 2^32).

1. R goes through a left shift of 4 and then is added to K[0]

2. R is added to Delta

3. R goes through a right shift of 5 and then is added to K[1]

An XOR operation is then applied to the result of those three operations and finally, the result of

the XOR operation is added to L. This result then becomes R for the next feistel round, because

of the swap.

**EQUVIVALENT KEY ATTACK PROCESS:**

We will implement the TEA which is block cipher. TEA is a variant of the Fiestel cipher, it is designed for speed and simplicity as a main aim, during the encryption of data this tea algorithm will mainly operate on 64 bit data which will be divided into two halves of 32 bit The key size used in TEA is 128bits, and a magic constant is also utilized which is defined as 2^32/(the golden ratio), which equals 2654435769. This constant is added to itself during each round, and is used to prevent attacks that try to take advantage of symmetry between rounds.

This attack will be implemented on 1st round of tea in which the key length is reduced to 64 bits because  size is effectivelyas opposed to the 128bits that would normally be utilized in 64 rounds of TEA. This is because of the fiestel nature of TEA; only half of the 128bit key is used in the first round. The remaining 64 bits of the key are introduced in the 2nd round, but because we are attacking 1 round of TEA we will focus on only half of the 128bit key.  Using known plaintext and their resulting ciphertext, we will perform a brute force attack on the 64bit key by repeatedly guessing 32bits of the key, which will eventually lead us to deduce the other 32bits.

TEA has a 128 bit key that is split up into four 32 bit subkeys, which can be seen as K[0],k[1],k[2],k[3] in thediagram. Delta is defined as a constant, 2^32/(golden ratio), which is 2654435769 as an integer. Multiples of delta are used in each round (mod 2^32).

In the first feistel round R is used as an input to several operations. All addition operations are (mod 2^32).

1. R goes through a left shift of 4 and then is added to K[0]

2. R is added to Delta

3. R goes through a right shift of 5 and then is added to K[1]

An XOR operation is then applied to the result of those three operations and finally, the result of the XOR operation is added to L. This result then becomes R for the next feistel round, because of the swap.

**Description of the partial key search attack:**

We implemented our attack on 1 Feistal round of TEA. In 1 Feistal round of TEA the key size is effectively reduced to 64 bits, which we call the first 32 bits subkey T and the second 32 bits subkey U. Normally a 128 bit key would be used in 2 or more rounds of TEA. This is because of the Feistal nature of TEA; only half of the 128 bit key is used in the first Feistal round. The remaining 64 bits of the key are used in the 2nd round. Using known plaintext and ciphertext pairs, we will perform a brute force attack on the 64 bit bit key by starting from 0 and incrementing by one until we guess a correct value for subkey T. This means that our brute force search will be of the order 2^32. We can do this because of our ability to calculate subkey U once we guess a value for T.

We are able to calculate U because of our ability to solve the encryption formula for subkey U. In the diagram above you can see the steps required to solve the TEA cipher equation for subkey U. The TEA cipher operations are substituted for F(R0, K1), and after some subtracting and XORing on either side of the equation we are end up with an equation solved for subkey U. Because L0, R0, and R1 are known (from the plaintext/ciphertext file generated using the oracle), and Delta is also known, all we have to do is guess a value for subkey T in order to solve for U. Additionally, if we use a different set of L0, R0, and R1 with the same Delta and the same T, we should get the same U. This “different set” of L0, R0, and R1 is really just a different or 2nd plaintext/ciphertext pair that we need to use. If we test a 2nd plaintext/ciphertext pair and we get a different value for U, that means that our guess for subkey T was incorrect, and we can increment our guess and do the calculation again. Thus, we can continually do this until we find a value for T that gives the same value for U for all of the plaintexts and ciphertexts.

**DESCRIPTION OF THE PARTIAL KEY SEARCH ATTACK**.

Our attack on 1 Feistal round of TEA is implemented. In 1 Feistal round of TEA the key size is effectively reduced to 64 bits, which we call the first 32 bits subkey T and the second 32 bits subkey U. Normally a 128 bit key would be used in 2 or more rounds of TEA. This is because of the Feistal nature of TEA; only half of the 128 bit key is used in the first Feistal round. The remaining 64 bits of the key are used in the 2nd round. Using known plaintext and ciphertext pairs, we will perform a brute force attack on the 64 bit bit key by starting from 0 and incrementing by one until we guess a correct value for subkey T. This means that our brute force search will be of the order 2^32. We can do this because of our ability to calculate subkey U once we guess a value for T.

We are able to calculate U because of our ability to solve the encryption formula for subkey U. In the diagram above you can see the steps required to solve the TEA cipher equation for subkey U. The TEA cipher operations are substituted for F(R0, K1), and after some subtracting and XORing on either side of the equation we are end up with an equation solved for subkey U. Because L0, R0, and R1 are known (from the plaintext/ciphertext file generated using the oracle), and Delta is also known, all we have to do is guess a value for subkey T in order to solve for U. Additionally, if we use a different set of L0, R0, and R1 with the same Delta and the same T, we should get the same U. This “different set” of L0, R0, and R1 is really just a different or 2nd plaintext/ciphertext pair that we need to use. If we test a 2nd plaintext/ciphertext pair and we get a different value for U, that means that our guess for subkey T was incorrect, and we can increment our guess and do the calculation again. Thus, we can continually do this until we find a value for T that gives the same value for U for all of the plaintexts and ciphertexts.

**METHOD FOR DETERMINING U AFTER T IS FOUND IS :**

L1=r0

R1=L0+f(R0,K1)

R1=L0+(((R0<<4+T)XOR((R0>>5)+U)XOR(R0+DELTA1))

R1-L0=(((R0<<4+T)XOR((R0>>5)+U)XOR(R0+DELTA1))

(RI-L0)XOR((R0<<4)+t)= ((R0>>5)+U)XOR(R0+DELTA1))

(RI-L0)XOR((R0<<4)+t)XOR(R0+DELTA1)= (R0>>5)+U

U=(RI-L0)XOR((R0<<4)+t)XOR(R0+DELTA1)-(R0>>5)

**DESCRIPTION OF ATTACK PROGRAM**

Our attack program takes one argument which is the name of the file with the plaintext/ciphertext pairs generated by the Oracle program. The plaintext/ciphertext pairs are in the following format:

[64 bit plaintext, 64 bit ciphertext]

The general idea of the attack program is that we only have to bruteforce 2^32 possible values. This is because given a subkey T, we can calculate a value for subkey U by using known plaintext/ciphertext pairs.  The following steps/pseudocode outline the general procedure of the attack program.

1. Read in the file with the plaintext/ciphertext pairs into lists. These values are then converted from strings in the file to 32 bit unsigned integers.

2. Guess a value for subkey T, starting at 0. 3. Calculate the value for U using our guess and the first plaintext/ciphertext pair 4. Calculate the value for U using our guess and the second plaintext/ciphertext pair 5. Do those two values of U match? 6. If not, this is the incorrect guess for T. Increment our guess and go back to Step 3. 6. If yes, we need to verify that this guess for subkey T is correct 7. Repeat steps 3 and 4 ten times with different plaintext/ciphertext pairs 8. If the values of U did not match every time, increment our guess for T and go back to Step 3. 9. If the values of U matched every time, this is the correct guess for T and we’ve found the key!!

An interesting note about our attack methodology was the number of plaintexts/ciphertexts required for the verifying step . We initially had this set at 100 instead of 10, but we decided to reduce it to see if the attack would still succeed.

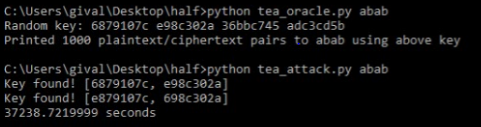
**ANALYSIS:**

Our attack program was successful in finding  random encryption keys that were generated through the use of our oracle program. Our oracle program selects a random 128 bit

key, and then generates 100 random plaintext/ciphertext pairs using 1 Feistal round of TEA encryption. These plaintext/ciphertext pairs are what the attack program uses to calculate the key. This is what the output of the oracle looks like for one of the keys that we generated.

Keep in mind that because we are only attacking 1 Feistal round of TEA, we are only trying to discover the first 64 bits of the key because the other 64 bits of the key are not used in the 1st round of TEA.

**Attack1:**

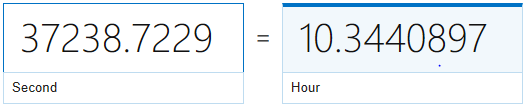


**DESCRIPTION OF THE ATTACK**

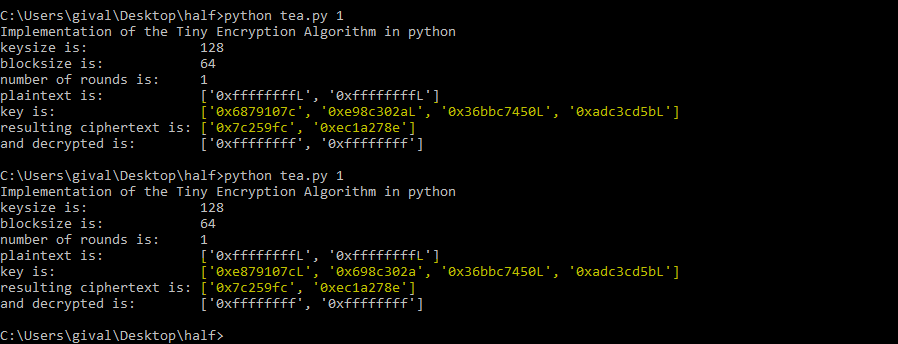
**Random key** :is the key which is used by oracle program in order to develop those pain/cipher text pairs which was further displayed in the file named abab

**Key found!** : this is the key found after our attack was successful the first key found was a part of the random Key whereas second one is the equvivalent key that what we are searching for

**Seconds:** This seconds shows the figure of seconds that were taken in order to check or obtain this equvivalent key (that is 37238.721999 sec in our attack which is nearly equals to nearly 10.35 hrs)



**To prove that these keys are actually equivalent:**



You can understand the behaviour of the equivalent key by observing the text that is being highlighted in the above screenshot (yellow coloured) here you can observe that the change in key even produced the same cipher text and which is the main function and equivalent key and this property of having the equivalent keys will affect the performance of the algorithm that is the security of the algorithm…

This attack was performed by running script on Laptop with following specifications:

OS: Windows 10,64 bit operating system,X64-based processor

Processor: Intel® Core™ i3-6th Gen @ 2.00Ghz clock speed

RAM: 8.00GB

**LINEAR CRYPTANALYSIS:**

In this section, we outline the approach to attacking a cipher using linear cryptanalysis based on the example cipher of our basic SPN.

**OVERVIEW OF BASIC ATTACK**

Linear cryptanalysis tries to take advantage of high probability occurrences of linear expressions involving plaintext bits, "ciphertext" bits (actually we shall use bits from the 2nd last round output), and subkey bits. It is a known plaintext attack: that is, it is premised on the attacker having information on a set of plaintexts and the corresponding ciphertexts. However, the attacker has no way to select which plaintexts (and corresponding ciphertexts) are available. In many applications and scenarios it is reasonable to assume that the attacker has knowledge of a random set of plaintexts and the corresponding ciphertexts.

The basic idea is to approximate the operation of a portion of the cipher with an expression that is linear where the linearity refers to a mod-2 bit-wise operation (i.e., exclusive-OR denoted by "⊕"). Such an expression is of the form:

 (1)

where Xi represents the i-th bit of the input X = [X1, X2, ...] and Yj represents the j-th bit of the output Y = [Y1, Y2, ...]. This equation is representing the exclusive-OR "sum" of u input bits and v output bits.

The approach in linear cryptanalysis is to determine expressions of the form above which have a high or low probability of occurrence. (No obvious linearity such as above should hold for all input and output values or the cipher would be trivially weak.) If a cipher displays a tendency for equation (1) to hold with high probability or not hold with high probability, this is evidence of the cipher’s poor randomization abilities. Consider that if we randomly selected values for u + v bits and placed them into the equation above, the probability that the expression would hold would be exactly 1/2. It is the deviation or bias from the probability of 1/2 for an expression to hold that is exploited in linear cryptanalysis: the further away that a linear expression is from holding with a probability of 1/2, the better the cryptanalyst is able to apply linear cryptanalysis. In the remainder of the paper, we refer to the amount by which the probability of a linear expression holding deviates from 1/2 as the linear probability bias. Hence, if the expression above holds with probability pL for randomly chosen plaintexts and the corresponding ciphertexts, then the probability bias is pL – 1/2. The higher the magnitude of the probability bias, |pL – 1/2|, the better the applicability of linear cryptanalysis with fewer known plaintexts required in the attack.

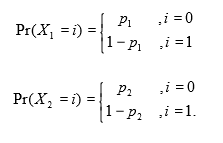
There are several ways to mount the attack of linear cryptanalysis. In this paper, we shall focus on what Matsui calls Algorithm 2 [1]. We investigate the construction of a linear approximation involving plaintext bits as represented by X in (1) and the input to the last round of the cipher (or equivalently the output of the 2nd last round of the cipher) as represented by Y in (1). The plaintext bits are random and consequently so are the input bits to the last round. Equation (1) could be equivalently reformulated to have the right side being the sum of a number of subkey bits. However, in (1) as written with the right side of "0", the equation implicitly has subkey bits involved: these bits are fixed but unknown (as they are determined by the key under attack) and implicity absorbed into the "0" on the right side of equation (1) and the probability pL that the linear expression holds. If the sum of the involved subkey bits is "0", the bias of (1) will have the same sign (+ or −) as the bias of the expression involving the subkey sum and, if the sum of the involved subkey bits is "1", the bias of (1) will have the opposite sign. Note that pL = 1 implies that linear expression (1) is a perfect representation of the cipher behaviour and the cipher has a catastrophic weakness. If pL = 0, then (1) represents an affine relationship in the cipher, also an indication of a catastrophic weakness. For mod-2 addition systems, an affine function is simply the complement of a linear function. Both linear and affine approximations, indicated by pL > 1/2 and pL < 1/2, respectively, are equally susceptible to linear cryptanalysis and we shall generally use the term linear to refer to both linear and affine relationships.

The natural question to ask is: How do we construct expressions which are highly linear and, hence, can be exploited? This is done by considering the properties of the cipher’s only nonlinear component: the S-box. When the nonlinearity properties of the S-box are enumerated, it is possible to develop linear approximations between sets of input and output bits in the S-box. Consequently, it is possible to concatenate linear approximations of the S-boxes together so that intermediate bits (i.e., data bits from within the cipher) can be cancelled out and we are left with a linear expression which has a large bias and involves only plaintext and the last round input bits.

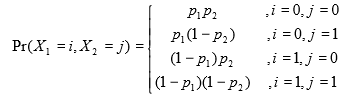
**PILING-UP PRINCIPLE**

Before we consider constructing a linear expression for the example cipher of this paper, we need some basic tools. Consider two random binary variables, X1 and X2. We begin by noting the simple relationships: X1⊕X2 = 0 is a linear expression and is equivalent to X1 = X2; X1⊕X2 = 1 is an affine expression and is equivalent to 2 1 XX ≠ .

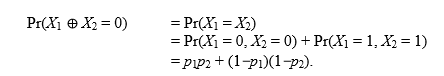
Now, assume that the probability distributions are given by



If the two random variables are independent, then



and it can be shown that



Another perspective is to let p1 = 1/2+ε1 and p2 = 1/2+ε2, where ε1 and ε2 are the probability biases and −1/2 ≤ ε1,ε2 ≤ +1/2. Hence, it follows that



and the bias ε1,2 of X1 ⊕ X2 = 0 is



This can be extended to more than two random binary variables, X1 to Xn, with probabilities p1 = 1/2+ε1 to pn = 1/2+εn. The probability that X1 ⊕ ... ⊕ Xn = 0 holds can be determined by the Piling-Up Lemma which assumes that all n random binary variables are independent.

Piling-Up Lemma (Matsui [1]) For n independent, random binary variables, X1, X2, ...Xn,



or, equivalently,



where ε1,2,..,n represents the bias of X1 ⊕ ... ⊕ Xn = 0.

Note that if pi = 0 or 1 for all i, then Pr(X1 ⊕ ... ⊕ Xn = 0) = 0 or 1. If only one pi = 1/2, then Pr(X1 ⊕ ... ⊕ Xn = 0) = 1/2.

In developing the linear approximation of a cipher, the Xi values will actually represent linear approximations of the S-boxes. For example, consider four independent random binary variables, X1, X2, X3 and X4. Let Pr(X1 ⊕ X2 = 0) = 1/2 + ε1,2 and Pr(X2 ⊕ X3 = 0) = 1/2 + ε2,3 and consider the sum X1 ⊕ X3 to be derived by adding X1 ⊕ X2 and X2 ⊕ X3 together. Hence,



So we are combining linear expressions to form a new linear expression. Since we may consider random variables X1 ⊕ X2 and X2 ⊕ X3 to be independent, we can use the PilingUp Lemma, to determine



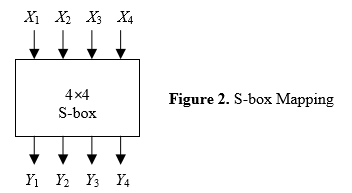
and, consequently,

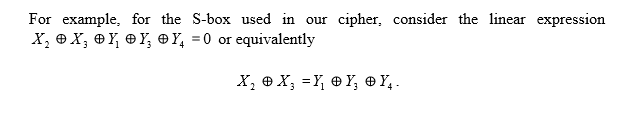


As we shall see, the expressions X1 ⊕ X2 = 0 and X2 ⊕ X3 = 0 are analogous to linear approximations of S-boxes and X1 ⊕ X3 = 0 is analogous to a cipher approximation where the intermediate bit X2 is eliminated. Of course, the real analysis will be more complex involving many S-box approximations.

**ANALYZING THE CIPHER COMPONENTS**

Before considering the attack in any more detail on the overall cipher, we first require knowledge of the linear vulnerabilities of an S-box. Consider the S-box representation of Figure 2 with input X = [X1 X2 X3 X4] and a corresponding output Y = [Y1 Y2 Y3 Y4]. All linear approximations can be examined to determine their usefulness by computing the probability bias for each. Hence, we are examining all expressions of the form of equation (1) where X and Y are the S-box input and outputs, respectively.





Applying all 16 possible input values for X and examining the corresponding output values Y, it may be observed that for exactly 12 out the 16 cases, the expression above holds true. Hence, the probability bias is 12/16−1/2 = 1/4. This is presented in Table 3.

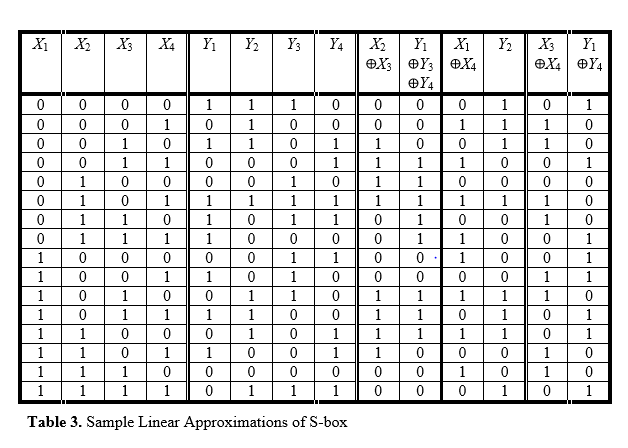
Similarly, for equation

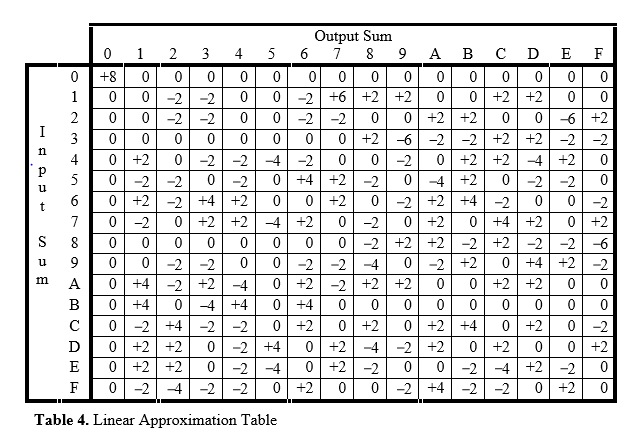


it may be seen that the probability bias is 0 and for equation



the probability bias is 2/16−1/2 = −3/8. In the last case, the best approximation is an affine approximation as indicated by the minus sign. However, the success of the attack is based on magnitude of the bias and, as we shall see, affine approximations can be used equivalently to linear approximations.





A complete enumeration of all linear approximations of the S-box in our cipher is given in the linear approximation table of Table 4. Each element in the table represents the number of matches between the linear equation represented in hexadecimal as "Input Sum" and the sum of the output bits represented in hexadecimal as "Output Sum" minus 8. Hence, dividing an element value by 16 gives the probability bias for the particular linear combination of input and output bits. The hexadecimal value representing a sum, when viewed as a binary value indicates the variables involved in the sum. For a linear combination of input variables represented as a1⋅X1⊕a2⋅X2⊕a3⋅X3⊕a4⋅X4 where ai ∈ {0,1} and "⋅" represents binary AND, the hexadecimal value represents the binary value a1a2a3a4, where a1 is the most significant bit. Similarly, for a linear combination of output bits b1⋅Y1 ⊕ b2⋅Y2 ⊕ b3⋅Y3 ⊕ b4⋅Y4 where bi ∈ {0,1}, the hexadecimal value represents the binary vector b1b2b3b4. Hence, the bias of linear equation X3 ⊕ X4 = Y1 ⊕ Y4 (hex input 3 and hex output 9) is −6/16 = −3/8 and the probability that the linear equation holds true is given by 1/2 − 3/8 = 1/8.

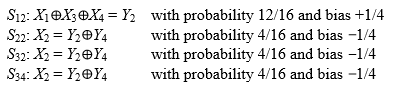
Some basic properties of the linear approximation table can be noted. For example, the probability that any sum of a non-empty subset of output bits is equal to the sum involving no input bits is exactly 1/2 since any linear combination of output bits must have an equal number of zeros and ones for a bijective S-box. Also, the linear combination involving no output bits will always equal the linear combination of no input bits resulting in a bias of +1/2 and a table value of +8 in the top left corner. Hence, the top row of the table is all zeros, except for the leftmost value. Similarly, the first column is all zeros except for the topmost value. It can also be noted the sum of any row or any column must be either +8 or −8. We leave the proof of this as an exercise to the reader.

**CONSTRUCTING LINEAR APPROXIMATIONS FOR THE COMPLETE CIPHER**

Once the linear approximation information has been compiled for the S-boxes in an SPN, we have the data to proceed with determining linear approximations of the overall cipher of the form of equation (1). This can be achieved by concatenating appropriate linear approximations of S-boxes. By constructing a linear approximation involving plaintext bits and data bits from the output of the second last round of S-boxes, it is possible to attack the cipher by recovering a subset of the subkey bits that follow the last round. We illustrate with an example.

Consider an approximation involving S12, S22, S32, and S34 as illustrated in Figure 3. Note that this actually develops an expression for the first 3 rounds of the cipher and not the full 4 rounds. We shall see how this is useful in deriving the subkey bits after the last round in the next section.

We use the following approximations of the S-box:



Letting Ui (Vi) represent the 16-bit block of bits at the input (output) of the round i Sboxes and Ui,j (Vi,j) represent the j-th bit of block Ui (Vi) (where bits are numbered from 1 to 16 from left to right in the figure of the cipher). Similarly, let Ki represent the subkey block of bits exclusive-ORed at the input to round i, with the exception that K5 is the key exclusive-ORed at the output of round 4.

Hence, U1 = P ⊕ K1 where P represents the block of 16 plaintext bits and "⊕" represents the bit-wise exclusive-OR. Using the linear approximation of the 1st round, we then have



with probability 3/4. For the approximation in the 2nd round, we have



with probability 1/4. Since U2,6 = V1,6 ⊕ K2,6, we can get an approximation of the form



with probability 1/4 and combining this with (2) which holds with probability of 3/4 gives

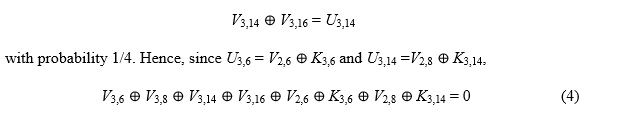


which holds with probability of 1/2 + 2(3/4−1/2)(1/4−1/2) = 3/8 (that is, with a bias of −1/8) by application of the Piling-Up Lemma. Note that we are using the assumption that the approximations of S-boxes are independent which, although not strictly correct, works well in practice for most ciphers.

For round 3, we note that



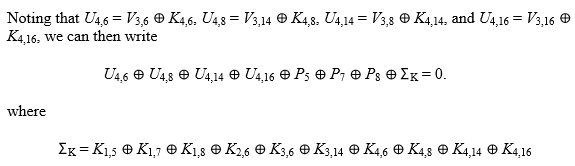
with probability 1/4 and



with probability of 1/2 + 2(1/4−1/2)2 = 5/8 (that is, with a bias of +1/8). Again, we have applied the Piling-Up Lemma.

Now combining (3) and (4), to incorporate all four S-box approximations, we get





and ΣK is fixed at either 0 or 1 depending on the key of the cipher. By application of the Piling-Up Lemma, the above expression holds with probability 1/2+23(3/4−1/2)(1/4−1/2)3 = 15/32 (that is, with a bias of −1/32).

Now since ΣK is fixed, we note that



must hold with a probability of either 15/32 or (1−15/32) = 17/32, depending on whether ΣK = 0 or 1, respectively. In other words, we now have a linear approximation of the first three rounds of the cipher with a bias of magnitude 1/32. We must now discuss how such a bias can be used to determine some of the key bits.

**EXTRACTING KEY BITS**

Once an R−1 round linear approximation is discovered for a cipher of R rounds with a suitably large enough linear probability bias, it is conceivable to attack the cipher by recovering bits of the last subkey. In the case of our example cipher, it is possible to extract bits from subkey K5 given a 3 round linear approximation. We shall refer to the bits to be recovered from the last subkey as the target partial subkey. Specifically, the target partial subkey bits are the bits from the last subkey associated with the S-boxes in the last round influenced by the data bits involved in the linear approximation.

The process followed involves partially decrypting the last round of the cipher. Specifically, for all possible values of the target partial subkey, the corresponding ciphertext bits are exclusive-ORed with the bits of the target partial subkey and the result is run backwards through the corresponding S-boxes. This is done for all known plaintext/ciphertext samples and a count is kept for each value of the target partial subkey. The count for a particular target partial subkey value is incremented when the linear expression holds true for the bits into the last round’s S-boxes (determined by the partial decryption) and the known plaintext bits. The target partial subkey value which has the count which differs the greatest from half the number of plaintext/ciphertext samples is assumed to represent the correct values of the target partial subkey bits. This works because it is assumed that the correct partial subkey value will result in the linear approximation holding with a probability significantly different from 1/2. (Whether it is above or below 1/2 depends on whether a linear or affine expression is the best approximation and this depends on the unknown values of the subkey bits implicitly involved in the linear expression.) An incorrect subkey is assumed to result in a relatively random guess at the bits entering the S-boxes of the last round and as a result, the linear expression will hold with a probability close to 1/2.

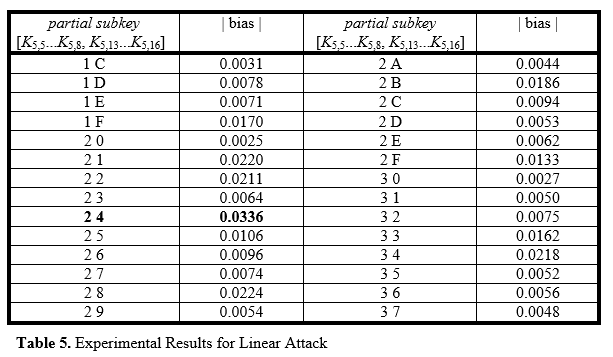
Let’s now put this into the context of our example. The linear expression of (5) affects the inputs to S-boxes S42 and S44 in the last round. For each plaintext/ciphertext sample, we would try all 256 values for the target partial subkey [K5,5...K5,8, K5,13...K5,16]. For each partial subkey value, we would increment the count whenever equation (5) holds true, where we determine the value of [U4,5...U4,8, U4,13...U4,16] by running the data backwards through the target partial subkey and S-boxes S24 and S44. The count which deviates the largest from half of the number of plaintext/ciphertext samples is assumed to the correct value. Whether the deviation is positive or negative will depend on the values of the subkey bits involved in ΣK. When ΣK = 0, the linear approximation of (5) will serve as the estimate (with probability < 1/2) and when ΣK = 1, (5) will hold with a probability > 1/2.

We have simulated attacking our basic cipher by generating 10000 known plaintext/ciphertext values and following the cryptanalytic process described for partial subkey values [K5,5...K5,8] = [0010] (hex 2) and [K5,13...K5,16] = [0100] (hex 4). As expected, the count which differed the most from 5000 corresponded to target partial subkey value [2,4]hex, confirming that the attack has successfully derived the subkey bits. Table 5 highlights a partial summary of the data derived from the subkey counts. (The complete data involves 256 data entries, one for each target partial subkey value.) The values in the table indicate the bias magnitude derived from

| bias | = | count − 5000 | / 10000

where the count is the count corresponding to the particular partial subkey value.

As can be seen from the partial results in the table, the largest bias occurs for partial subkey value [K5,5...K5,8, K5,13...K5,16] = [2,4] and this observation was, in fact, found to be true for the complete set of partial subkey values.



The experimentally determined bias value of 0.0336 is very close to the expected value of 1/32 = 0.03125. Note that, although the correct target partial subkey has clearly the highest bias, other large bias values occur indicating that the examination of incorrect target partial subkeys is not precisely equivalent to comparing random data to a linear expression (where the bias could be expected to be very close to zero). Inconsistencies in the experimental biases can occur for several reasons including the S-box properties influencing the partial decryption for different partial subkey values, the imprecision of the independence assumption required for use in the Piling-Up Lemma, and the influence of linear hulls (to be discussed in the next section).

**COMPLEXITY OF ATTACK**

We refer to the S-boxes involved in the linear approximation as active S-boxes. In Figure 3, the four S-boxes in rounds 1 to 3 influenced by the highlighted lines are active. The probability that a linear expression holds true is related to the linear probability bias in the active S-boxes and the number of active S-boxes. In general, the larger the magnitude of the bias in the S-boxes, the larger the magnitude of the bias of the overall expression. Also, the fewer active S-boxes, the larger the magnitude of the overall linear expression bias.

Let ε represent the bias from 1/2 of the probability that the linear expression for the complete cipher holds. In his paper, Matsui shows that the number of known plaintexts required in the attack is proportional to ε−2 and, letting NL represent the number of known plaintexts required, it is reasonable to approximate NL by



In practice, it is generally reasonable to expect some small multiple of ε−2 known plaintexts are required. Although strictly speaking, the complexity of the cryptanalysis could be characterized in both time and space (or memory) domains, we refer to the data required to mount the attack when considering the complexity of the cryptanalysis. That is, we assume that if we are able to acquire NL plaintexts, we are able to process them.

Since the bias is derived using the Piling-Up Lemma where each term in the product refers to an S-box approximation, it is easy to see that the bias is dependent on the biases of the S-box linear approximations and the number of active S-boxes involved. General approaches to providing security against linear cryptanalysis have focused on optimizing the S-boxes (i.e., minimizing the largest bias) and finding structures to maximize the number of active S-boxes. The design principles of Rijndael are an excellent example of such an approach.

It must be cautioned, however, the concept of a "proof" of security to linear cryptanalysis is usually premised on the nonexistence of highly likely linear approximations. However, the computation of the probability of such linear approximations is based on the assumption that each S-box approximation is independent (so that the Piling-Up Lemma can be used) and on the assumption that one linear approximation scenario (i.e., a particular set of active S-boxes) is sufficient to determine the best linear expression between plaintext bits and data bits at the input to the last round. The reality is that the Sbox approximations are not independent and this can have significant impact on the computation of the probability. Also, linear approximation scenarios involving the same plaintext and last round input bits but different sets of active S-boxes can combine to give a linear probability higher than that predicted by one set of active S-boxes. This concept is referred to as a linear hull [16]. Most notably for example, a number of linear approximation scenarios may have very small biases and on their own seem to imply that a cipher might be immune to a linear attack. However, when these scenarios are combined, the resulting linear expression of plaintext and last round input bits might have a very high bias. Nevertheless, the approach outlined in this paper, tends to work well for many ciphers because the independence assumption is a reasonable approximation and when one linear approximation scenario of a particular set of active S-boxes has a high bias, it tends to dominate the linear hull.