CRYPTOGRAPHY

Cluster Innovation Centre
University of Delhi

Akshit Jain (11905)

Anshika (11909)

Bhavya Tewari (11913)

Bhavya Verma (11914)

Gaurav Dubey (11919)

Shivam Goyal (11942)

Yashu Garg (11949)

April 2020

Month Long Project submitted for the paper
Art of communication and Creative Writing

Certificate of Originality

The work embodied in this report entitled **“Cryptography”** has been carried out by **Akshit Jain, Anshika, Bhavya Tewari, Bhavya Verma, Gaurav Dubey, Shivam Goyal, Yashu Garg** for the paper “**Art of communication and Creative Writing**”. We declare that the work and language included in this project report is free from any kind of plagiarism.

AKSHIT JAIN

ANSHIKA

BHAVYA TEWARI

BHAVYA VERMA

GAURAV DUBEY

SHIVAM GOYAL

YASHU GARG

Acknowledgement

Primarily, we would like to thank God for being able to complete this project with success. Then we would like to thank our teachers, Prof. Dorje Dawa, whose valuable guidance has been the ones that helped us patch this project and make it a success, their instructions, and suggestions have served as the major contributor toward the completion of the project.

Then we would like to thank our parents and friends who have helped us with their valuable suggestions and guidance has been helpful in various phases of this project.

Last but not the least, we would like to thank our fellow classmates, without whose help this would not have been possible.

Table of Contents

1. Abstract

Digital communication has become an essential part of infrastructure in the present world and it has witnessed a noticeable and continuous development in a lot of applications during last few decades. Nowadays, almost all applications are Internet-based and it is important that communication be made confidential and secure. It is then important to devise methods that secure data from misuse or mishandling of any kind.

Cryptography is one of the most important and widely used techniques that are used to provide information security over the open and insecure networks such as Internet. Cryptography distorts the original message to ensure that encrypted data or resource is not made available to an undesired user for any purpose.

This paper aims to study the concept of cryptography and various techniques by which cryptography is practiced. The paper also attempts to give the idea of how modern-day cryptography has been carved out from older practices of encryption.

2. Introduction

Cryptography, as the name suggests, means “hidden” and means “writing”, i.e. is {method of protecting information and communication} with the use of coding so that only those for whom the information is intended can read and process it.

In today’s era, due to tremendous growth of networking technologies an enormous amount of data is being exchanged over the Internet as a result of which security of information being conveyed over the Internet is becoming more significant as sensitive data needs to be transferred securely over the internet while maintaining its confidentiality, integrity, usability availability. *[1] **

To maintain the privacy and security of confidential and sensitive information there is a need of approaches which enhances the level of information security. Information hiding is one of the many available approaches which increase the level of information security. The most powerful and widely used approaches of information hiding used to contravene the threats to information security are Cryptography and Steganography *[1] **. Cryptography provides security by manipulating the original confidential information so that it is not readable to any intruder and no information can be processed using the encountered encryption.

For this purpose, a data undergoes a process to get itself secured by a key on the encryption end and it can be used only when it reaches the decryption key for retrieval of data. A simple string example can be used to gain clarity of the mentioned idea. Consider a string “GREEK”. The encryption algorithm for the string gives out a number which has number of digits equal to double the size of the string. The number has alphabetical positions of the last digit first with the process going up to the first letter. The resultant encrypted code will be “1105051807”.

Cryptography is used nowadays in almost all applications that use Internet as means of communication. Real time applications of cryptography include ATM machines; password protection of email passwords, social account (Facebook, twitter, etc.) passwords; E-commerce, Defence forces; intelligent agencies. *[2] **.

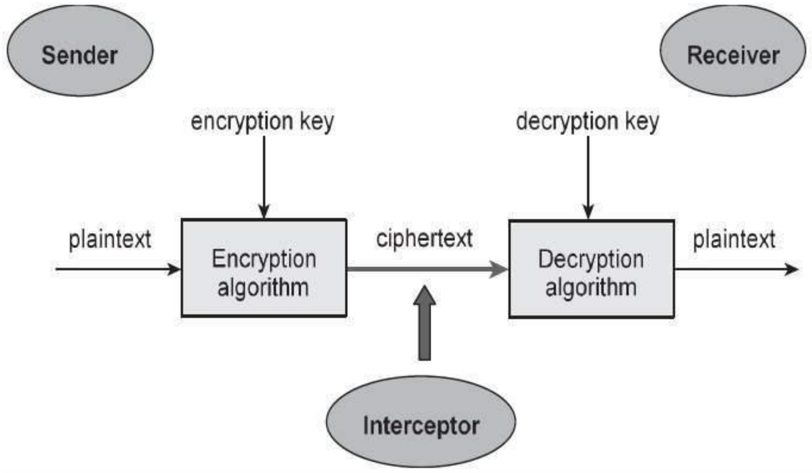


Fig 2.1 Schematic depiction of a cryptographic function

To understand and analyse any cryptographic technique we need to have an idea of Ciphers and Cryptanalysis. Cipher is an algorithm which is used for encryption and decryption i.e. a set of rules that an information has to go through to get itself converted into a form that is readable by a decryption key. Cryptanalysis is the method of studying, understanding, analysing and testing a cipher or any enciphering system so as to know ways for strengthening or weakening it. It gives the measure of efficiency of any ciphering algorithm and possible was to improve a weak one.

3. Classic Cryptography

Earlier, cryptography was mainly focused on converting messages into unreadable figures in order to protect it during the time it was being carried from one place to another. Early cryptography was mainly focused on converting messages into unreadable figures in order to protect it during the time it was being carried from one place to another. They used digit or letter of the inputs. They were usually symmetric key techniques.

Some of the cryptographic techniques that have been into usage have been discussed below.

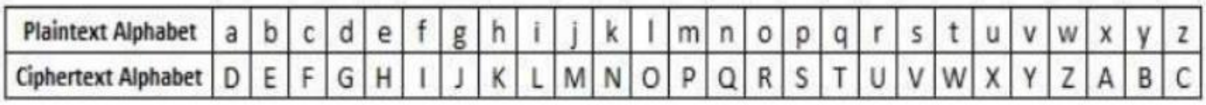
3.1. Monoalphabetic and Polyalphabetic Cipher

- **Monoalphabetic cipher** is a substitution cipher in which for a given key, the cipher alphabet for each plain alphabet is fixed throughout the encryption process. For example, if ‘A’ is encrypted as ‘D’, for any number of occurrences in that plaintext, ‘A’ will always get encrypted to ‘D’. These ciphers are highly susceptible to cryptanalysis.
- **Polyalphabetic Cipher** is a substitution cipher in which the cipher alphabet for the plain alphabet may be different at different places during the encryption process.

3.2. Caesar Cipher

It is a mono-alphabetic cipher wherein each letter of the plaintext is substituted by another letter to form the ciphertext. It is a simplest form of substitution cipher scheme. This cryptosystem is generally referred to as the **Shift Cipher**. The concept is to replace each alphabet by another alphabet which is ‘shifted’ by some fixed number between 0 and 25.

For this type of scheme, both sender and receiver agree on a ‘secret shift number’ for shifting the alphabet. This number which is between 0 and 25 becomes the key of encryption.



Plaintext Alphabet	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext Alphabet	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

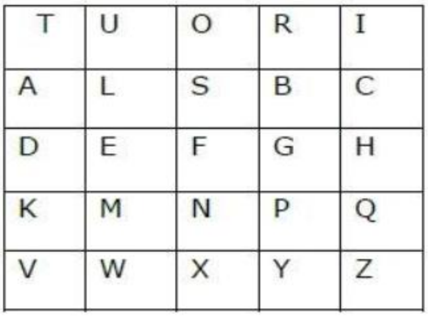
Fig 3.2.1 Caser Encipherment

3.3 Playfair Cipher

In this scheme, pairs of letters are encrypted, instead of single letters as in the case of simple substitution cipher.

In Playfair cipher, initially a key table is created. The key table is a 5×5 grid of alphabets that acts as the key for encrypting the plaintext. Each of the 25 alphabets must be unique and one letter of the alphabet (usually J) is omitted from the table as we need only 25 alphabets instead of 26. If the plaintext contains J, then it is replaced by I.———

The sender and the receiver deicide on a particular key, say ‘tutorials. In a key table, the first characters (going left to right) in the table is the phrase, excluding the duplicate letters. The rest of the table will be filled with the remaining letters of the alphabet, in natural order. The key table works out to be –



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

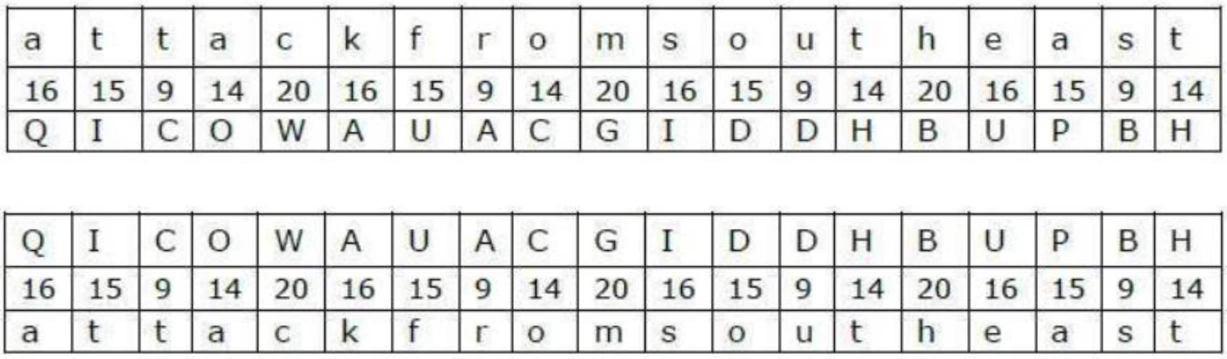
Fig 3.3.1 Playfair Cipher Table

3.4. Vigenere Cipher

This scheme of cipher uses a text string (say, a word) as a key, which is then used for doing a number of shifts on the plaintext. For example, let’s assume the key is ‘point’. Each alphabet of the key is converted to its respective numeric value: In this case,

p → 16, o → 15, i → 9, n → 14, and t → 20.

Thus, the key is: 16 15 9 14 20.



a	t	t	a	c	k	f	r	o	m	s	o	u	t	h	e	a	s	t
16	15	9	14	20	16	15	9	14	20	16	15	9	14	20	16	15	9	14
Q	I	C	O	W	A	U	A	C	G	I	D	D	H	B	U	P	B	H

Q	I	C	O	W	A	U	A	C	G	I	D	D	H	B	U	P	B	H
16	15	9	14	20	16	15	9	14	20	16	15	9	14	20	16	15	9	14
a	t	t	a	c	k	f	r	o	m	s	o	u	t	h	e	a	s	t

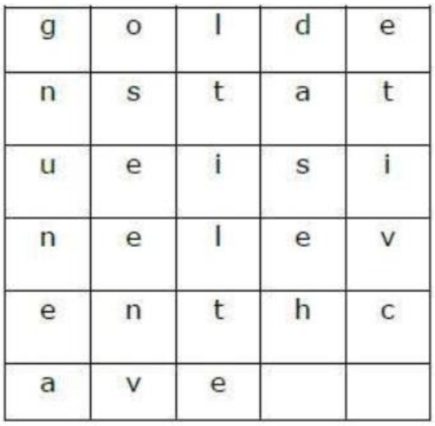
Fig 3.4.1 Cipher Text and Decryption key for Vigenere Cipher

3.5 Transposition Cipher

It is another type of cipher where the order of the alphabets in the plaintext is rearranged to create the ciphertext. The actual plaintext alphabets are not replaced.

An example is a ‘simple columnar transposition’ cipher where the plaintext is written horizontally with a certain alphabet width. Then the ciphertext is read vertically as shown.

For example, the plaintext is “golden statue is in eleventh cave” and the secret random key chosen is “five”. We arrange this text horizontally in table with number of column equal to key value. The resulting text is shown below.



g	o	l	d	e
n	s	t	a	t
u	e	i	s	i
n	e	l	e	v
e	n	t	h	c
a	v	e		

Fig 3.5.1 Transposition Cipher

4. Modern Cryptography

Modern cryptographic practices have been developed to weed out the possible inefficiencies that classic practices brought with them and are more secure and strong against any attempt to mishandle data. Unlike classic cryptography, which used digits and letters of inputs, it operates on binary bit sequences.

It relies on publicly known mathematical algorithms for coding the information. Secrecy is obtained through a secrete key which is used as the seed for the algorithms. The computational difficulty of algorithms, absence of secret key, etc., make it impossible for an attacker to obtain the original information even if he knows the algorithm used for coding.

Modern cryptography requires parties interested in secure communication to possess the secret key only.

5. Types of Cryptography

In general, there are four types of cryptography:

5.1 Symmetric Key Cryptography

“It is also called as single key cryptography. It uses a single key. In this encryption process the receiver and the sender has to agree upon a single secret(shared) key. Given a message (called plaintext) and the key, encryption produces unintelligible data, which is about the same length as the plaintext was. Decryption is the reverse of encryption, and uses same key as encryption *[4] **.

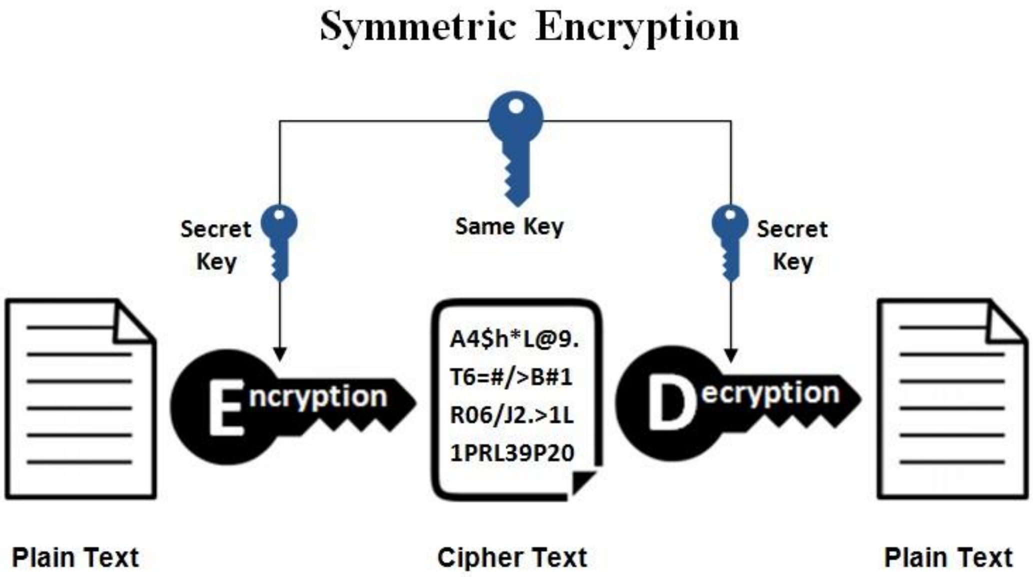


Figure 5.1.1 Symmetric Cryptography

The process of encryption consists of running a plaintext (input) through an encryption algorithm called a cipher, which in turn generates a ciphertext (output).

*While symmetric encryption offers a wide range of benefits, there is one major disadvantage associated with it: the inherent problem of transmitting the keys used to encrypt and decrypt data [5] *.* When these keys are shared over an unsecured connection, they are vulnerable to being intercepted by malicious third parties. If an unauthorized user gains access to a particular symmetric key, the security of any data encrypted using that key is compromised.

To solve this problem, many web protocols use a combination of symmetric and asymmetric encryption to establish secure connections. Among the most prominent examples of such a hybrid system is the Transport Layer Security (TLS) cryptographic protocol used to secure large portions of the modern internet.

5.1.1 Symmetric Key Algorithms

There are 2 types of Symmetric Key Algorithms:

5.1.1.1 Block Algorithms

Set lengths of bits are encrypted in blocks of electronic data with the use of a specific secret key. As the data is being encrypted, the system holds the data in its memory as it waits for complete blocks.

5.1.1.2 Stream Algorithms

Data is encrypted as it streams instead of being retained in the system’s memory.

5.1.2 Symmetric Key Encryption Techniques

"*Various algorithms have been developed so far to describe symmetric key* *cryptography. These are AES, DES, 3DES, Blowfish."[7] *.*

Brief definitions of the most common symmetric encryption techniques are given as follows:

5.1.2.1 DES (Data Encryption Standard)

DES was the first encryption standard to be recommended by NIST (National Institute of Standards and Technology). DES is (64 bits key size with 64 bits block size). Since that time, many attacks and methods recorded the weaknesses of DES, which made it an insecure block cipher.

5.1.2.2 3DES

3DES is an enhancement of DES. It is 64-bit block size with 192 bits key size. In this standard the encryption method is similar to the one in the original DES but applied 3 times to increase the encryption level and the average safe time.

5.1.2.3 RC2

RC2 is a 64-bits block cipher with a variable key size that range from 8 to 128 bits. RC2 is vulnerable to a related-key attack using 234 chosen plaintexts.

5.1.2.4 Blowfish

Blowfish is block cipher 64-bit block - can be used as a replacement for the DES algorithm. It takes a variable-length key, ranging from 32 bits to 448 bits;

default 128 bits. Blowfish is unpatented, license-free, and is available free for all uses. Blowfish has variants of 14 rounds or less. Blowfish is successor to Two fish.

5.1.2.5 AES

AEF is a block cipher. It has variable key length of 128, 192, or 256 bits; default 256. it encrypts data blocks of 128 bits in 10, 12 and 14 rounds depending on the key size. AES encryption is fast and flexible; it can be implemented on various platforms especially in small devices. Also, AES has been carefully tested for many security applications.

5.1.2.6 RC6

RC6 is block cipher derived from RC5. It was designed to meet the requirements of the Advanced Encryption Standard competition. RC6 proper has a block size of 128 bits and supports key sizes of 128, 192 and 256 bits. Some references consider RC6 as Advanced Encryption Standard.

5.1.3 Uses of Symmetric Encryption

While symmetric encryption is an older method of encryption, it is faster and more efficient than asymmetric encryption, which takes a toll on networks due to performance issues with data size and heavy CPU use. Due to the better performance and faster speed of symmetric encryption (compared to asymmetric), symmetric cryptography is typically used for bulk encryption / encrypting large amounts of data, e.g. for database encryption. In the case of a database, the secret key might only be available to the database itself to encrypt or decrypt.

Some examples of where symmetric cryptography are used are:

- Payment applications, such as card transactions where PII needs to be protected to prevent identity theft or fraudulent charges.

- Validations to confirm that the sender of a message is who he claims to be.

- Random number generation or hashing.

5.1.4 Advantages of Symmetric Key Cryptography

- A symmetric cryptosystem is faster.

- In Symmetric Cryptosystems, encrypted data can be transferred on the link even if there is a possibility that the data will be intercepted. If there is no key transmitted with the data, the chances of data being decrypted are null.

- A symmetric cryptosystem uses password authentication to prove the receiver's identity.

- A system only which possesses the secret key can decrypt a message.

5.1.5 Disadvantages of Symmetric Key Cryptography

- Symmetric cryptosystems have a problem of key transportation. The secret key is to be transmitted to the receiving system before the actual message is to be transmitted. Every means of electronic communication is insecure as it is impossible to guarantee that no one will be able to tap communication channels. So, the only secure way of exchanging keys would be exchanging them personally.

- Cannot provide digital signatures that cannot be repudiated.

5.1.6 Managing Symmetric Key Encryption

Symmetric encryption does come with its own drawbacks. Its weakest point is its aspects of key management, including:

5.1.6.1 Key Exhaustion

Symmetric Encryption suffers from behaviour where every use of a key ‘leaks’ some information that can potentially be used by an attacker to reconstruct the key. The defences against this behaviour include using a key hierarchy to ensure that master or key-encryption keys are not over-used and the appropriate rotation of keys that do encrypt volumes of data. To be tractable, both these solutions require competent key-management strategies as if (for example) a retired encryption key cannot be recovered the data is potentially lost.

5.1.6.2 Attribution Data

Unlike asymmetric (public-key) *Certificates*, symmetric keys do not have embedded metadata to record information such as expiry date or an Access Control List to indicate the use the key may be put to - to Encrypt but not Decrypt for example.

Maintaining large-scale symmetric encryption systems is a very challenging task. This is especially true when we want to achieve banking-grade security and auditability when the corporate or IT architecture is decentralized or geographically distributed.

5.2 Asymmetric Key Cryptography

Asymmetric key cryptography also known as Public–key (PK) cryptography introduces a new concept. The idea can be visualized, by making a slot in the safe box so that everyone can deposit a message (like a letter box). However, only the receiver can open the safe and look at its contents. This concept was proposed by Diffie and Hellman.

Public–key cryptography is based on the concept of separating the key used to encrypt a message from the one used to decrypt it. Anyone who wants to send a message to a party, e.g., Bob, can encrypt that message using Bob's public key but only Bob can decrypt the message using his private key. It is understood that the private key should be kept secret at all times and the public key is publicly available to everyone. Furthermore, it is impossible for anyone, except him, to derive the private key (at least to do so in any reasonable amount of time). *[6]* *.

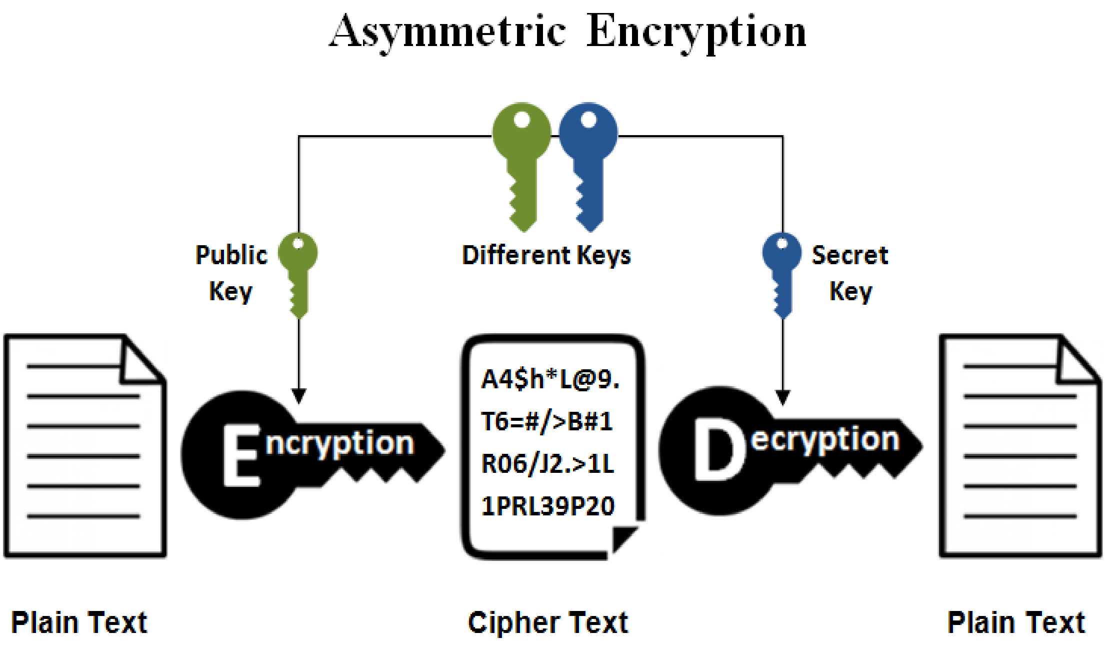


Fig. 5.2.1. Asymmetric Cryptography

5.2.1 How Asymmetric Cryptography Works

Asymmetric encryption uses a mathematically related pair of keys for encryption and decryption: a public key and a private key. If the public key is used for encryption, then the related private key is used for decryption; if the private key is used for encryption, then the related public key is used for decryption. The two participants in the asymmetric encryption workflow are the sender and the receiver; each has its own pair of public and private keys. First, the sender obtains the receiver's public key. Next, the plaintext or ordinary, readable text is encrypted by the sender using the receiver's public key; this creates ciphertext. The ciphertext is then sent to the receiver, who decrypts the ciphertext with their private key and returns it to legible plaintext. Because of the one-way nature of the encryption function, one sender is unable to read the messages of another sender, even though each has the public key of the receiver.

5.2.2 Asymmetric Key Encryption Algorithms

This section concentrates on two specific public–key algorithms, namely RSA and ECC. We show the mathematical background, the computational aspects, and some implementation numbers of each.

5.2.2.1 RSA

RSA is the most popular public–key algorithm and named after its creators Rivest, Shamir, and Adelman. Considering the public–key algorithms it is also the easiest to understand and implement. RSA was patented until 2000 and is today free for use. RSA can be used for encryption and, thus, for key transport and digital signature. In the following we introduce the RSA algorithm consisting of encryption/ decryption function and vulnerabilities *[7]* * .

➢ RSA Encryption

RSA is a block cipher mechanism. So, we divide the input binary text into 8 bits apart. We will convert the first 8-bit text into an integer form. After that we take a public key from key generator and perform encryption operation for that integer. For example, 'M' is an integer then we encrypt 'M' by performing C = Pe mod n [c is cipher text and e is public exponent]. After calculating the value of C, we will convert C into binary format. After that we will make binary value of C as 16-bit length and print that result in cipher txt. Now we will take another 8-bit text and repeat the above process.

➢ RSA Decryption

Divide the input binary text into 16 bits apart. We have converted the first 16-bit text into an integer form. After that we take a private key 'd' from key generator and perform decryption operation for that integer. For example, ‘C’ is an integer then we encrypt ‘C’ by performing P = Cd mod n.

➢ Vulnerabilities

RSA private keys are likely to be weak if their value is less than $N^{0.292}$. It is believed that for secure implementation private exponent to be larger than $N^{0.5}$. When RSA is implemented with several key pairs, the implementer often chooses to use the same N for all key pairs, thus saving computation time. However, since the private and public exponents together always assist in factoring N, every single member of the system will be able to factor N with his key pair and use that result to invert any public exponent to the corresponding private exponent. So, it is necessary to generate a new N value foreach key pair.

5.2.2.2 ECC

Elliptic Curve Cryptosystem is a relatively new cryptosystem, suggested independently in 1987 by Koblitz and in 1986 by Miller. ECC can be used instead of Diffie–Hellman and other DL-based algorithms. Elliptic curve cryptosystem is based on the discrete logarithm problem (DLP). The basic operation that needs to be performed for ECC is Q = k ·P, where k is an integer and P is a point of a finite group. This operation is known as scalar multiplication. Hence, we have to add the point P k-times to itself to get the solution *[6] **.

- **ECC Encryption**

The easiest way to do public key encryption with ECC is to use ECIES. In this system, the person doing the decryption has a private key aa (which is an integer) and a public key AA. He publishes his public key AA to everyone, and keeps his private key secret. Now, when any other person wants to pass a note to him, he first picks a random value bb, and computes the points bb and bAbA; he then gives the point bAbA to a key derivation function hh, which produces a set of symmetric keys; he then uses the symmetric keys to encrypt the message. He then sends the values and Encrypt the note to him.

- **ECC Decryption**

When He receives these two values, he first computes the point a(b)a(b), which is the same as b(a)=bAb(a)=bA; he then passes that point to the same key derivation function, which produces the same symmetric keys that he had. He then decrypts the value of the encrypted note recovering the note. If you examine this, you can see what they are effectively doing is performing an ECC operation, and then using the shared secret to encrypt a message. This might seem like we're cheating a bit, however this meets the criteria for public key encryption (anyone with the public key can encrypt, only the holder of the private key can decrypt).

- **Vulnerabilities**

  A secure implementation of the ECC curve is theoretically possible, it is not easy to achieve. In fact, incorrect implementations can lead to ECC private key leaks in a number of scenarios. There are numerous examples of how failed implementation of ECC algorithms resulted in significant vulnerabilities in the cryptographic software. A great example is that of the Sony ECDSA security disaster. Furthermore, there are examples of improper implementation of ECC in OpenSSL that resulted in common vulnerabilities. These vulnerabilities range from omission of the server key exchange message to malformed signatures.

5.2.2.3 Conclusion

After a short introduction to public–key cryptography, we concentrated on the engineering aspects of the primitives RSA and ECC. RSA is the most widely used asymmetric algorithm and ECC is especially promising for embedded application, due to the short key length compared to RSA. We summarized the mathematical background of these algorithms and introduced the state-of-the-art techniques necessary to implement them efficiently. We also give the reader an insight into the best published performance numbers of these algorithms implemented on different platforms, namely embedded microprocessors, general processors, FPGAs, and ASICs.

5.2.3 Advantages of asymmetric cryptography

The advantages of asymmetric cryptography include:

- The key distribution problem is eliminated because there's no need for exchanging keys.
- Security is increased as the private keys don't ever have to be transmitted or revealed to anyone.

The use of digital signatures is enabled so that a recipient can verify that a message comes from a particular sender.

- It allows for non-repudiation so the sender can't deny sending a message.

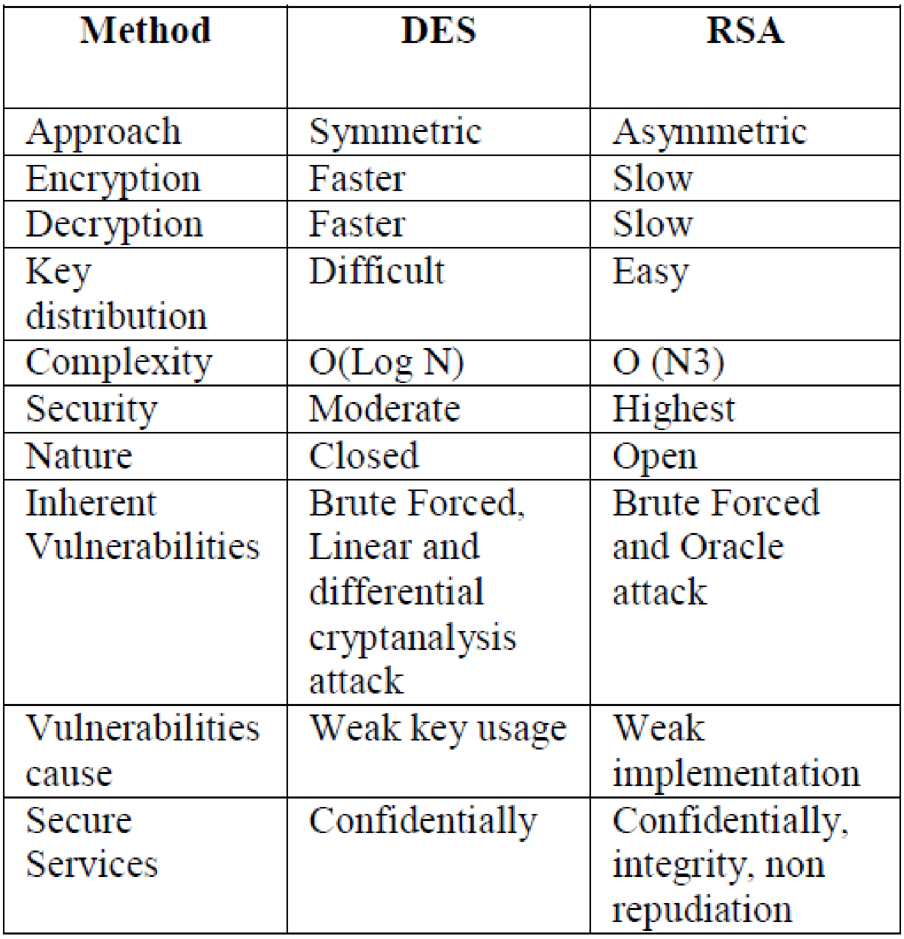
5.2.4 Disadvantages of asymmetric cryptography

The benefits of asymmetric cryptography include:

- It's a slow process compared to symmetric cryptography, so it's not appropriate for decrypting bulk messages.
- If an individual loses his private key, he can't decrypt the messages he receives.
- Since the public keys aren't authenticated, no one really knows if a public key belongs to the person specified. Consequently, users must verify that their public keys belong to them.
- If a hacker identifies a person's private key, the attacker can read all of that individual's messages.

5.2.5 Symmetric vs Asymmetric comparison

The main difference between these two methods of encryption is that asymmetric encryption algorithms makes use of two different but related keys -- one key to encrypt the data and another key to decrypt it -- while symmetric encryption uses the same key to perform both the encryption and decryption functions.



Method	DES	RSA
Approach	Symmetric	Asymmetric
Encryption	Faster	Slow
Decryption	Faster	Slow
Key distribution	Difficult	Easy
Complexity	O(Log N)	O (N3)
Security	Moderate	Highest
Nature	Closed	Open
Inherent Vulnerabilities	Brute Forced, Linear and differential cryptanalysis attack	Brute Forced and Oracle attack
Vulnerabilities cause	Weak key usage	Weak implementation
Secure Services	Confidentially	Confidentially, integrity, non repudiation

Table 5.2.5.1 DES vs RSA

5.3 Hash Functions

“Hash functions are functions that compress an input of arbitrary length to a result with a fixed length. They were introduced in cryptology in the late seventies as a tool to protect the authenticity of information. Soon it became clear that they were a very useful building block to solve other security problems in telecommunication and computer networks. If hash functions satisfy additional requirements, they are a very powerful tool in the design of techniques to protect the authenticity of information *[9] **.

“As they play an important role in ensuring the security and confidentiality of information, identification and authentication methods are approached with an ever-greater attention, both in civilian (personal information, passwords, PIN codes) and military domains. Practical applications of cryptographic hash functions include message integrity checking, digital signatures, authentication procedures and other information security related applications.

5.3.1 Hash Function Type

Cryptographic hash functions may be divided into two groups:

- keyed hash functions – require a secret key and are known as message authentication code (MAC),
- un-keyed hash functions – do not require any secret key and may be referred to as manipulation detection code (MDC).

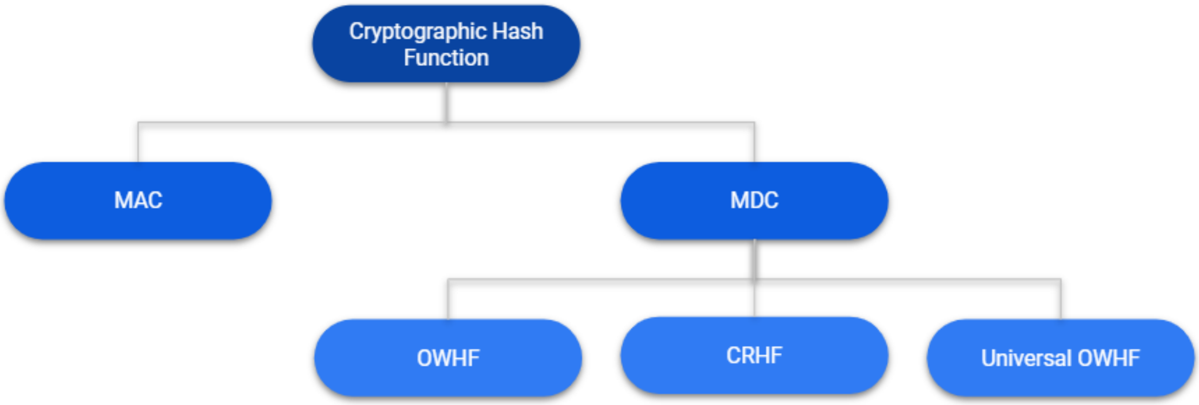


Figure 5.2.1. A taxonomy for cryptographic hash functions.

Generally, the term hash functions refer to un-keyed hash functions.

We will focus on un-keyed hash functions which can be divided into three subgroups, based on their additional properties:

1. One-way hash functions (OWHF) – defined by Merkle and fulfilling the following requirements:

   - ➢ hash function does not give any constraint on input data size,
   - ➢ output hash has constant length,
   - ➢ output hash should be easy to compute,
   - ➢ "given h and h(x), it is computationally infeasible to determine x" – a preimage resistance feature,
   - ➢ "given h and x, it is computationally infeasible to find an x0 6=x such feature that h(x) =h(x0)" – the second preimage resistance.

2. Collision resistant hash functions (CRHF) – belonging to the OWHF group and fulfilling an additional requirement: it is impossible to find a pair (x, x0) where x6=x0, which have the same hash value (h(x) = h(x0)). This condition is known as collision resistance. The difference between the second preimage resistance depends on the selection of arguments. In the second preimage resistance condition, the attacker has a given value x and has to find x0. In the collision resistance condition, the selection of both: x and x0 is a free choice of the attacker.

3. Universal one-way hash functions – a family in which the probability of finding a second preimage for a randomly chosen hash function is negligible. These functions are faster than CRHF and allow to omit trapdoors during digital signature creation. They are used when it is impossible to make a decision in which the hash function should be chosen before computation starts *[9] **.

5.3.2 Applications of hash functions

Cryptographic hash functions can be used to protect information authenticity and to protect against the threat of repudiation.

Hash functions can be used for identification with passwords, and help derive an encryption algorithm.

5.3.2.1 Information authentication

The basic idea of the use of cryptographic hash functions is to reduce the protection of the authenticity of information of arbitrary length to the protection of the secrecy and/or authenticity of quantities of fixed length. First, a distinction will be made between protection of authentication with and without secrecy. The second option is whether the protection of authenticity will depend on the secrecy and authenticity of a key or on the authenticity of an information dependent hash code.

5.3.2.2 Non-repudiation

The technical term non-repudiation denotes a service whereby the recipient is given guarantee of the message's authenticity, in the sense that the recipient can

subsequently prove to a third party that the message is authentic even if its originator subsequently revoked it. There is an elegant solution based on trapdoor one-way permutations. The first practical proposal of a public-key system with digital signature capability is the RSA cryptosystem. Its security is based on the fact that it is “easy” to find two large primes, but “hard” to factor their product. Subsequently new schemes appeared, based on the other number theoretic problems like the discrete log problem. The complexity theoretic approach has resulted in provably secure digital signature schemes based on claw-free pairs of permutations, one-way permutations, and finally on one-way functions, which can be shown to be optimal. A remarkable evolution here is the intertwining in some schemes between the signature and the hashing operation.

Digital signature schemes based on the practical approach were further optimized but have not received too much attention.

5.3.2.3 Identification with passwords

An MDC can be used to generate from a password or a passphrase a publicly accessible (readable) password file: one stores the MDC corresponding to the password or passphrase, instead of the password itself. Subsequently one has to protect only the integrity of the password file. In most applications it should be infeasible to derive a valid password from an entry in the file, which implies that a OWHF is sufficient. This is in fact one of the few cases where only finding a first preimage should be hard. Historically this is probably the first application of one-way functions. If a passphrase of arbitrary size has to be compressed, one will need a one-way hash function. A related application is commitment to a string without revealing it.

5.3.2.4 Encryption algorithms based on hash functions

Without going into details, one can remark that every hash function can be used in several ways to produce an encryption algorithm. A first possibility is to use the hash function as the F-function in a Feistel cipher. The text input of the round can be used as the chaining variable of the hash function, and the key can go to the data input. An alternative is to have a fixed chaining variable and to concatenate the data to the key. Interesting theoretical results can be shown if the hash function is pseudo-random and if the round keys are independent. A second possibility is to construct a key stream generator with a mode where the output of the hash function is fed back to the input, or where the input is derived from a counter. In case of a MAC the design can even be simpler, as the use of a secret key is already part of the algorithm. It is certainly possible to design more efficient encryption algorithms from scratch, but this type of solution could be acceptable for applications where encryption is required occasionally.

5.3.2.5 Application to software protection

MAC, MDC, and a digital signature scheme, the use of these three techniques can be applied to protect the integrity of software. The two parties involved in the application are the software vendor (who is also supposed to be the author of the software) and the user. The attacker will try to modify the software: this can be a computer virus, a competitor or even one of the parties involved. For this application there is clearly no need for secrecy.

Cryptographic hash functions provide an efficient way to protect integrity and to speed up digital signatures *[10] **.

5.4. Visual Cryptography

Visual cryptography is a cryptographic technique which allows visual information (pictures, text, etc.) to be encrypted in such a way that the decrypted information appears as a visual image.

One of the best-known techniques has been credited to Moni Naor and Adi Shamir, who developed it in 1994 *[11] **. They demonstrated a visual secret sharing scheme, where an image was broken up into n shares so that only someone with all n shares could decrypt the image, while any n − 1 shares revealed no information about the original image. Each share was printed on a separate transparency, and decryption was performed by overlaying the shares. When all n shares were overlaid, the original image would appear. There are several generalizations of the basic scheme including k-out-of-n visual cryptography, and using opaque sheets but illuminating them by multiple sets of identical illumination patterns under the recording of only one single-pixel detector.

Using a similar idea, transparencies can be used to implement a one-time pad encryption, where one transparency is a shared random pad, and another transparency acts as the cipher text. Normally, there is an expansion of space requirement in visual cryptography. But if one of the two shares are structured recursively, the efficiency of visual cryptography can be increased to 100%.

In the upcoming sub sections, we’ll further apprehend and analyze the technique developed by Naor-Shamir duo. We’ll also study other technique dealing with the well-known halftone technology and single pixel imaging.

5.4.1. (2, N) Visual Cryptography Sharing Case

Sharing a secret with an arbitrary number of people N such that at least 2 of them are required to decode the secret is one form of the visual secret sharing scheme presented by Moni Naor and Adi Shamir in 1994. In this scheme we have a secret image which is encoded into N shares printed on transparencies. The shares appear random and contain no decipherable information about the underlying secret image, however if any 2 of the shares are stacked on top of one another the secret image becomes decipherable by the human eye.

Every pixel from the secret image is encoded into multiple sub pixels in each share image using a matrix to determine the color of the pixels. In the (2, N) case a white pixel in the secret image is encoded using a matrix from the following set, where each row gives the sub pixel pattern for one of the components:

$$\{\text{all permutations of the columns of}\} : C_0 = \begin{bmatrix} 1\ 0\ \ldots\ 0 \\ 1\ 0\ \ldots\ 0 \\ \ldots \\ 1\ 0\ \ldots\ 0 \end{bmatrix}$$

While a black pixel in the secret image is encoded using a matrix from the following set:

$$\{\text{all permutations of the columns of}\} : C_1 = \begin{bmatrix} 1\ 0\ \ldots\ 0 \\ 0\ 1\ \ldots\ 0 \\ \ldots \\ 0\ 0\ \ldots\ 1 \end{bmatrix}$$

For instance, in the (2, 2) sharing case (the secret is split into 2 shares and both shares are required to decode the secret) we use complementary matrices to share a black pixel and identical matrices to share a white pixel. Stacking the shares, we have all the sub pixels associated with the black pixel now black while 50% of the sub pixels associated with the white pixel remain white.

5.4.2. The Halftone Technology

Halftone is the reprographic technique that simulates continuous-tone imagery through the use of dots, varying either in size or in spacing, thus generating a gradient-like effect. "Halftone" can also be used to refer specifically to the image that is produced by this process *[12] **.

Where continuous-tone imagery contains an infinite range of colors or greys, the halftone process reduces visual reproductions to an image that is printed with only one color of ink, in dots of differing size (pulse-width modulation) or spacing (frequency modulation) or both. This reproduction relies on a basic optical illusion: when the halftone dots are small, the human eye interprets the patterned areas as if they were smooth tones. At a microscopic level, developed black-and-white photographic film also consists of only two colors, and not an infinite range of continuous tones.

Just as color photography evolved with the addition of filters and film layers, color printing is made possible by repeating the halftone process for each subtractive color – most commonly using what is called the "CMYK color model". The semi-opaque property of ink allows halftone dots of different colors to create another optical effect, full-color imagery.

Fig.5.4.2.1. shows how the halftone dots actually are arranged (left) and how the human eye would perceive it from a sufficient distance (right).

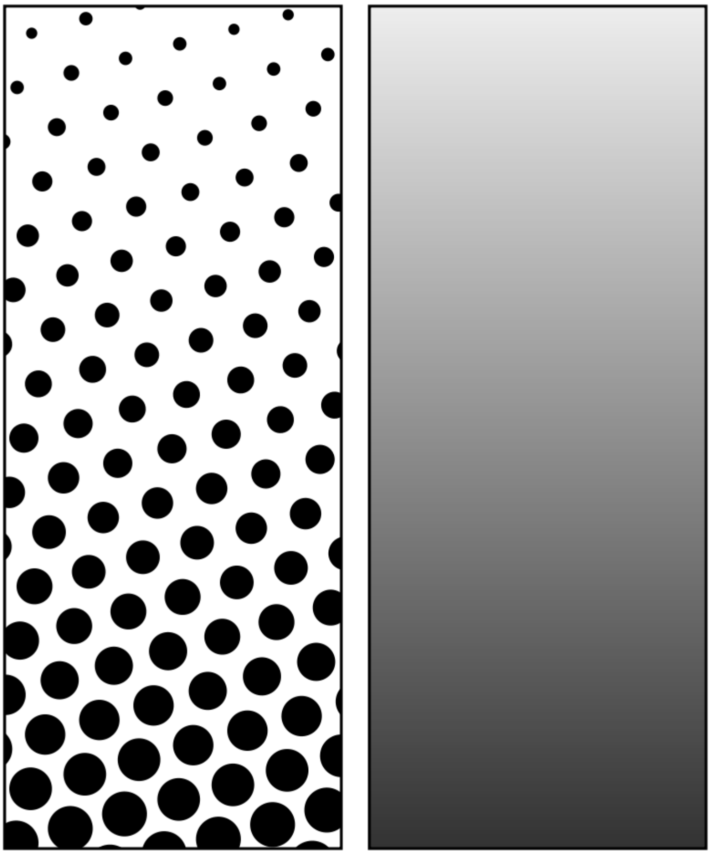


Fig. 5.4.2.1. Halftone dots arrangement.

5.4.3. Principles of single-pixel imaging (SPI)

Single-pixel imaging is an emerging paradigm that allows high-quality images to be provided by a device that is only equipped with a single point detector. Such a device is referred to as a single-pixel camera.*[13] **.

In SPI, the projection device will sequentially project N different illumination patterns $P_1(x, y)$, $P_2(x, y)$ … $P_n(x, y)$ onto the object image O(x, y). Then a sequence of single-pixel light intensities I_1, I_2, . . ., I_n is recorded by the single-pixel bucket detector. For the nth ($1 < n < N$) illumination pattern, In is mathematically the inner product between O (x, y) and $P_n(x, y)$, given

by Eq. (1).

$$I_n = \iint O(x, y) P_n(x, y)\, dx \qquad \ldots (1)$$

The object image O (x, y) can be reconstructed from all the illumination patterns $P_n(x, y)$ ($1<n < N$) and the single-pixel intensity sequence I_n ($1 < n < N$) by various methods. In this work, random binary illumination patterns are used and each pixel in $P_n(x, y)$ is randomly set to be 0 or 1. A typical optical setup for SPI is shown in Fig. 5.4.3.1.

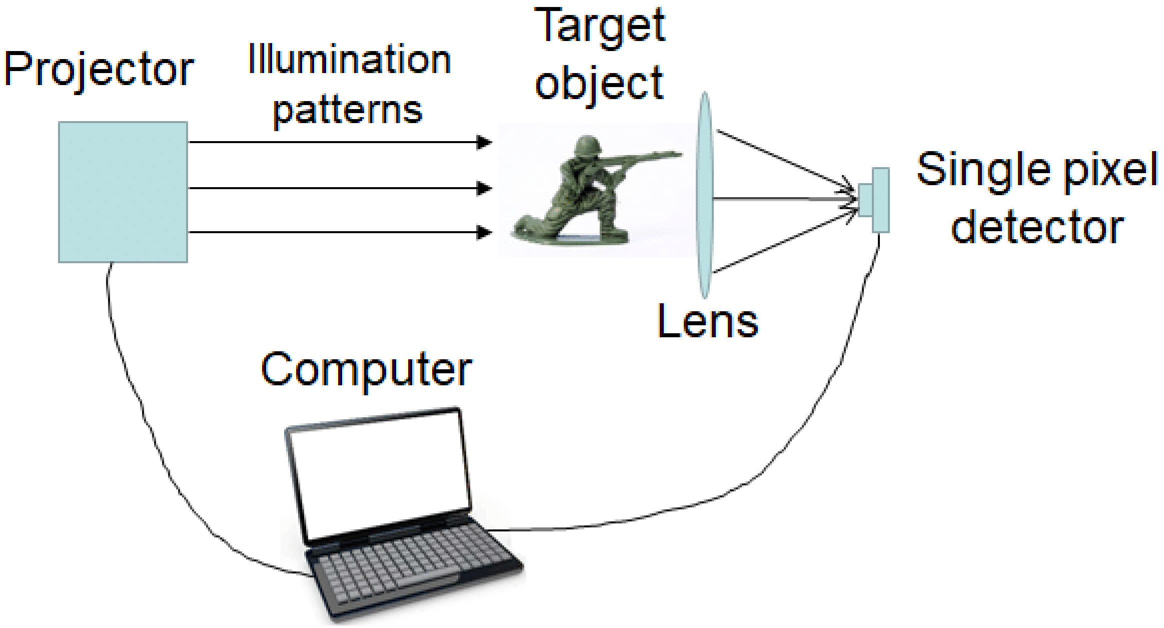


Fig. 5.4.3.1. Optical setup of a single-pixel imaging system.

6. Performance Analysis of Encryption Algorithm

6.1 Block Size comparison

In modern cryptography, ciphers are generally divided into stream ciphers and block ciphers. Block ciphers operate on a fixed length string of bits. The length of this bit string is the **block size**.

Algorithm	**BLOCK SIZE (bits)**
AES	128
DES	64
BLOWFISH	64
MD-5	512
SHA-1	
RSA	Minimum 512
RC5	Variable

Table 6.1. Block Size

6.2 Key Length Comparison

Key length is equal to the number of bits in an encryption algorithm's key. A short key length means poor security. The key length determines the maximum number of combinations required to break an encryption algorithm.

Algorithm	**KEY LENGTH (bits)**
AES	128/192/256
DES	56
BLOWFISH	128 to 448

MD-5	128
SHA-1	160
RSA	1024
RC5	Variable (up to 2048)

Table 2. Key Length Comparison

6.3 Encryption time

The average encryption time of the selected algorithms are compared in the figure below.The average time can be obtained by running each of the algorithms 50 times, recording the encryption time of each run and calculating the average time of each. Blowfish algorithm is the fast while RSA is the slowest *[3] **.

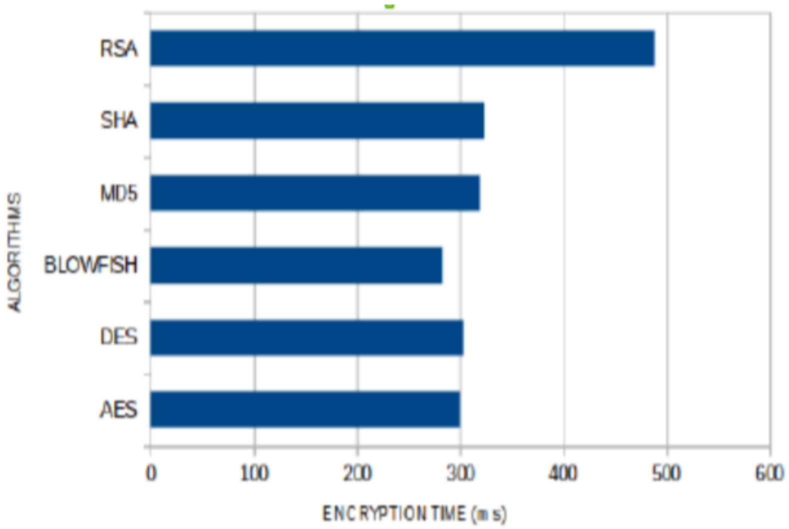


Fig 6.3.1. Encryption time comparison

6.4 Security

The security of the selected algorithms is compared in terms of the known attacks on encryption types. This shows the number of known attacks to break the selected algorithms.

Algorithm	**Known Attacks**
AES	Brute Force, XSL Attack, Known-key Distinguishing Attack or Biclique Attack, Side-Channel Attack
DES	Meet-in-the-Middle Attack, the Davies Attack, Differential cryptanalysis, Linear cryptanalysis, differential-linear cryptanalysis, Memory Tradeoff, Exhaustive Search, ne-round attack, Full 16 round Attack
BLOWFISH	Birthday attacks, Known-plaintext attacks, SWEET32 Attack
MD-5	Collision Attack
SHA-1	The SHAppening, SHAttered
RSA	Elementary Attacks e.g Coppersmith’s short pad attack, Franklin Reiter related message attack, Hastad’s Broadcast attack. Implementation attacks such as Timing attacks and Random Fault.

Table 6.3.1. Possible attacks

7. Conclusion

In this paper we have studied the idea and necessity of information security through Cryptography and related techniques. The first two sections explained integral concepts like encryption, decryption, ciphers and cryptanalysis- which are crucial to analyse a particular encryption algorithm. Classic encryption algorithms have also been precisely presented. This study allowed us to understand that classic algorithms vulnerable to security attacks thus calling for Modern encryption algorithms.

Classification in Modern Cryptography has been covered extensively. A thorough analysis has also been carried out on the following broad parameters- block size, key length, encryption time and security. The relevance of these parameters has also been justified. The comparative study showed that RSA has the longest encryption time with approximately 490ms, and blowfish with the least encryption time of approximately 290ms, RSA and MD5 have the largest block size of 512 bits while DES and blowfish have the least block size of 56 bits. RC 5 has the longest key length that makes a greater number of attacking combinations but considering the totality of the parameters BLOWFISH is the strongest of the selected algorithms. Advanced algorithms like Triple DES are better for more complex purposes.

Digital communication involves data transfer in visual form whose security is maintained by Visual cryptographic techniques. Digital signatures for protecting software and applications are secured efficiently by Hash Functions. The efficiency of encryption algorithms always has scope of improvement which can be worked upon on knowing all possible threats to them.

Hence, we have contemplated and analysed how information transaction can be provided proper security through the use of various cryptography techniques. After realizing the advantages and disadvantages of each, we ‘ve come to the conclusion that though there’s been tremendous enhancements in the cryptography scenario, there’s still scope for improvement.

References

[1] S. Almuhammadi and A. Al-Shaaby, “A Survey on Recent Approaches Combining Cryptography and Steganography”, Computer Science & Information Technology (CS & IT), 2017.

[2] Critical Analysis of Cryptography and Steganography, Alpa Agath*, Chintan Sidpara, Darshan Upadhyay,2018.

[3] A Comparative Study of Some Traditional and Modern Cryptographic Techniques Oguntunde, Arekete,Odim,and Olakanmi,Redeemer’s University Ede, NIGERIA, 2017.

[4] Chandra, S. P. (2014). A Comparative Survey Of Symmetric And Asymmetric Key Cryptography. 2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE).

[5] Katz, J. L. (2007). Introduction to Modern Cryptography. New York: CRC PRESS.

[6] Sandeep Kumar and Thomas Wollinger: Fundamentals of Symmetric. 2012 Horst Gortz Institute (HGI) for Security in Information Technology, Ruhr University of Bochum, Germany.

[7] Thakur, J. K. (2011). DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis. International Journal of Emerging Technology and Advanced Engineering.

[8] Preneel, B. (1994). Cryptographic hash functions. European Transactions on Telecommunications, 5(4), 431-448.

[9] Tchórzewski, J & Jakóbik, A. (2019). Theoretical and Experimental Analysis of Cryptographic Hash Functions. Journal of Telecommunications and Information Technology.

[10] Preneel, B. (1993). Analysis and design of cryptographic hash functions (Doctoral dissertation, Katholieke Universiteit te Leuven).

[11] Shamir A. Naor, M. Visual cryptography. Springer Berlinn Heidelberg, Berlin, Heidelberg, 1995.

[12] Young-Chang Hou, Department of Information Management, National Central University, 2002.

[13] Shuming Jiao, Jun Feng, Yang Gao, Ting Lei and Xiaocong Yuan, Visual cryptography in single-pixel imaging, 2020.