CRYPTOGRAPHIC HASH FUNCTIONS FOR IMAGE PROCESSING

by

Shafaq Iftikhar

B.S(ENG)., COMSATS Institute of IT, Pakistan, 2007

A project

presented to Ryerson University

in partial fulfillment of the

requirements for the degree of

Master of Engineering

in the Program of

Electrical and Computer Engineering

Toronto, Ontario, Canada, June 2015

©Shafaq Iftikhar 2015

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this MEng Project. This is a true copy of the MEng Project, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this MEng Project to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this MEng Project by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that MEng Project may be made electronically available to the public.

Cryptographic Hash Functions for Image Processing, M.Eng. 2015, Shafaq Iftikhar, Program of Electrical and Computer Engineering, Ryerson Universty

ABSTRACT

In this paper, a novel algorithm based on hash function for image cryptography is proposed. In this algorithm, the key idea is to encrypt half of the image using data from the second half of the image and then apply it to each other. This scheme can achieve high sensitivity, high complexity, and high security. The sole purpose is to improve the image entropy.

ACKNOWLEDGEMENTS

I am using this opportunity to express my gratitude to everyone who supported me throughout the course of this M.Eng project. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project. It gives me great pleasure in acknowledging my Professor Dr. Kaamran Raahemifar for the constant support and guidance for the success of this project.

I would like to thank my parents and family for being a great example and encouraging me throughout my studies, making this project possible.

Dedicated to my parents.

Contents

List of Figures

List of Tables

Chapter 1

1 INTRODUCTION

Plaintext is the information which is easily understandable without any difficulty or use of particular method. Encryption is a technique used for concealing the data in such a way that only the person intended for this information will be able to comprehend it. Such text is called cipher text. The method used for converting cipher text back to plain text is called decryption.

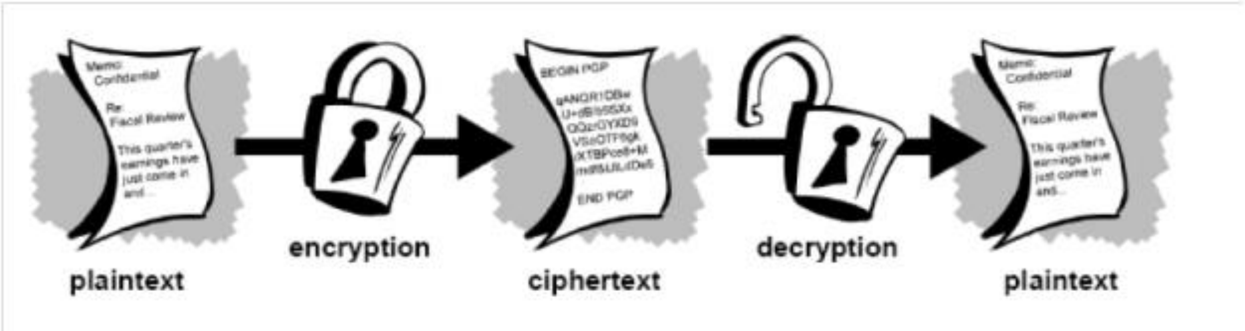


Figure 1: Encryption and Decryption

1.1 Background

1.1.1 Cryptography

Cryptanalysis is the science of analyze and break secure communication. Classical cryptanalysis involves an interesting combination of analytical reasoning, application of mathematical tools, pattern finding, patience, determination, and luck. Cryptanalysts are also called attackers. Cryptology embraces both cryptography and cryptanalysis.

Cryptography is the knowledge and use of mathematical techniques to encode and decode data. This technique assists in the transmission of data in a secured manner which is accessible in readable form only to the genuine recipient. Similar to Cryptography, there is another branch to these techniques called cryptanalysis. This is the art of having information security such as confidentiality, data integrity, entity authentication, and data origin authentication.

1.1.2 Cryptography Algorithm

A cryptographic method or cipher is a numerical function or procedure utilized as a part for encrypting and decryption procedure. A cryptographic method consists of a key, an expression, a number, or an expression used for encryption from the plaintext. This same plaintext can be encoded to variety of contents that have their own distinctive keys.

The protection of this encoded information is altogether focused on to two things: the quality of the cryptographic calculation and the mystery of the key. A cryptographic method, all potential keys and all the procedures and rules make it in to a cryptosystem.

1.1.3 Conventional Cryptography

In conventional cryptography, also called secret-key or symmetric-key encryption, one key is utilized to do both processes i-e encryption and decryption. The Data Encryption

Standard (DES) is one of the instances of a conventional cryptosystem commercially used even by the Federal Government. Figure 2 is an illustration of the conventional encryption process.

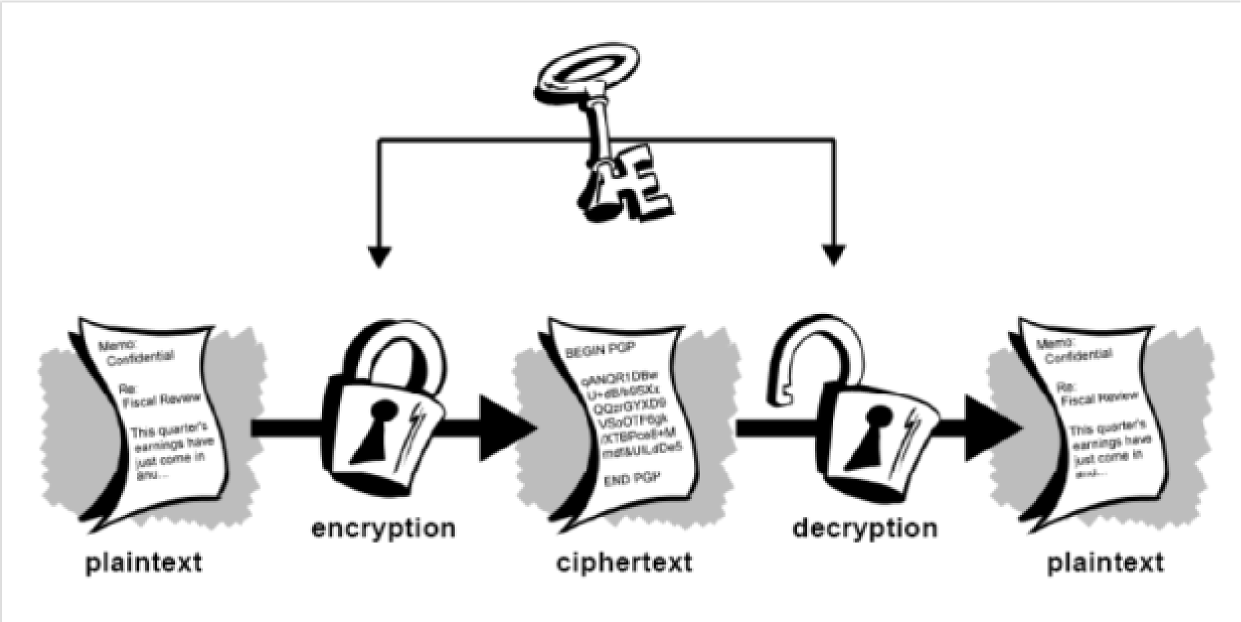


Figure 2: Conventional Encryption

Conventional encryption has its own positives. It is very fast and reliable, particularly when the encrypted data is not leaving. Nevertheless, conventional encryption for transmitting secure data by itself can be very pricey just because of the complicatedness and trouble of secure key sharing.

1.1.4 Instance

It will be a good example to recall a character from any favorite spy movie:

There is somebody with a sealed briefcase fastened to his or her wrist. So what is secured in the briefcase, anyhow? It’s undoubtedly not the missile launch code or bio-toxin formula or invasion plan itself.

In fact it’s the key that resolve decrypting the secret data. Secret key is the only approach for a sender and recipient to interconnect securely consuming conventional encryption, so they must come to an understanding upon a single key and keep it secret among themselves. In case both are present in different physical locations, they must have reliance on a courier, or any other protected communication medium to inhibit the revelation of the secret key throughout the transmission. Anyone who listen to or interrupts the key in transfer can later deliver, amend, and falsify all information encrypted or authenticated with that key. From DES to Captain Midnight’s Secret Decoder Ring, the determined problem with conventional encryption is key circulation where the important question arises that how the key can be given to recipient without having anyone’s interception in transmission.

1.1.5 Encryption Methods

1.1.5.1 DES

The Data Encryption Standard (DES) is a block cipher that practises mutual secret encryption. It was a selection by the National Bureau of Standards as an official Federal Information Processing Standard (FIPS) for the United States in 1976 and which has consequently appreciated extensive use internationally. It is constructed on a symmetric-key algorithm that utilizes a 56-bit key. The algorithm was in the beginning controversial with classified design elements, a comparatively short key length, and uncertainties about a National Security Agency (NSA) way out. DES subsequently came under passionate academic inspection which encouraged the modern appreciative of block ciphers and their cryptanalysis. This key size is susceptible to an instinctive force attack using current technology.

1.1.5.2 Triple DES

It is a variation of DES, Triple DES, which is responsible for considerably improved security by accomplishing the core DES algorithm three times in a row. The consequence of creating the DES encryption is much more challenging to instinctive force. Triple-DES is assessed to be 2 to the 56th times more tough to break than DES. Triple DES can still be well thought-out as a secure encryption algorithm. Triple DES is also written as 3-DES or 3DES.

1.1.5.3 AES

Advanced Encryption Standard is a symmetric cipher well-defined in Federal Information Processing (FIPS) Standard Number 197 in 2001 as the federal government sanctioned encryption algorithm. The NSA has permitted 128-bit AES for practise up to secret level and 192-bit AES for practise up to top secret level. AES is based upon the Rijndael algorithm, developed by Joan Daemen and Vincent Rijmen. AES identifies three permitted key lengths: 128-bits, 192-bits and 256-bits.

1.1.6 Public Key Cryptography

The difficulties of key dissemination are deciphered by public key cryptography which is an asymmetric scheme that customs a pair of keys for encryption: a public key for encrypting data, and a consistent private, or secret key for decryption. Public key is public to the world while protecting private key secret. Any person with a duplicate of public key can then encrypt information that only he or she can read. Even to people who have not ever met.

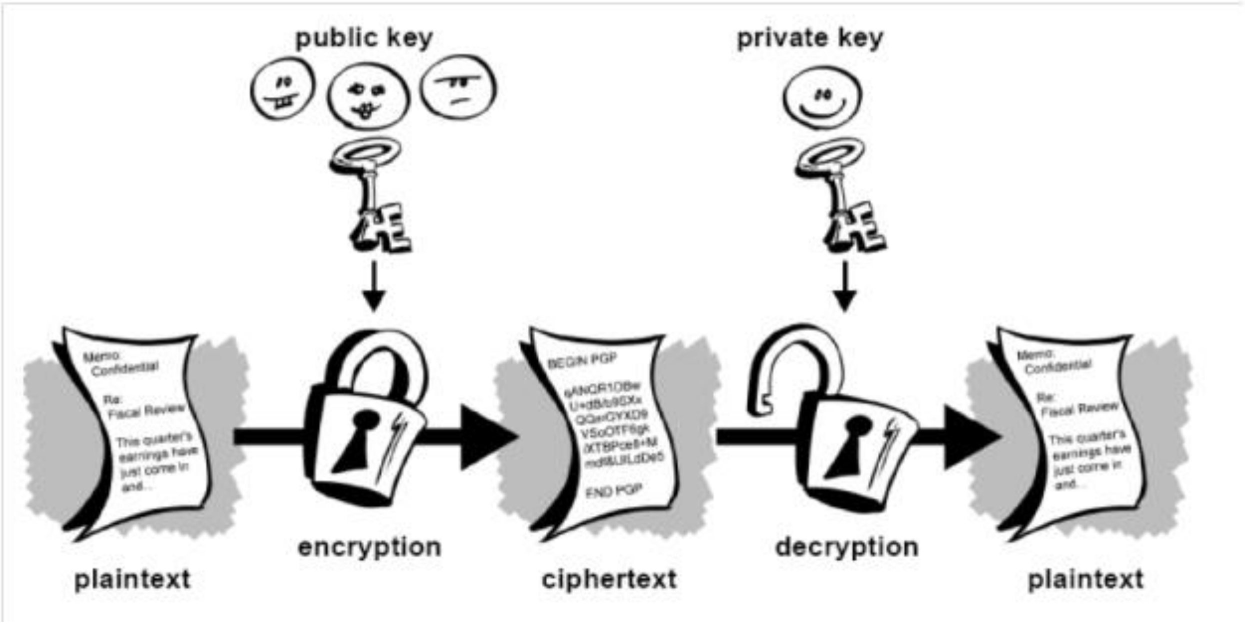


Figure 3: Public Key Encryption

The principal advantage of public key cryptography is that it permits general public who have no pre-existent security procedure to interchange messages securely. The requirement for sender and receiver to share secret keys through some secure channel is rejected; all communications consist of only public keys, and no private key is ever transferred or shared. Some examples of public-key cryptosystems are

- Elgamal (named for its inventor, Taher Elgamal),
- RSA (named for its inventors, Ron Rivest, Adi Shamir, and Leonard Adleman),
- Diffie-Hellman (named, you guessed it, for its inventors), and
- DSA, the Digital Signature Algorithm (invented by David Kravitz)

For the reason that conventional cryptography was once the only accessible resources for transmitting secret information, the outflow of secure channels and key distribution downgraded its use only to those who could meet the expense of it, such as governments and large banks. Public key encryption is the technological uprising that be responsible for sturdy cryptography to the mature multitudes. Recall the courier with the protected briefcase handcuffed to his or her wrist? Public-key encryption situates him out of business, perhaps to its liberation.

Some of the public key encryption algorithm are

1.1.6.1 RSA

It is an Internet encryption and verification system that practises an algorithm established in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman. The RSA algorithm is the utmost frequently used encryption and verification algorithm and is contained within the portion of the Web browsers from Microsoft and Netscape. It's also portion of Lotus Notes, Intuit's Quicken, and several other products. RSA security is the owner of the encryption system. The algorithm technologies are licensed by the company and the company also trades development kits. The technologies are part of current or anticipated Web, Internet, and computing standards.

1.1.6.2 Elliptic curve cryptography (ECC)

It is a methodology to public-key cryptography based on the algebraic structure of elliptic curves over finite fields. In 1985, Neal Koblitz and Victor S. Miller recommended the use of elliptic curves in cryptography independently. Elliptic curves are correspondingly used in numerous integer factorization algorithms that consume applications in cryptography, for example Lenstra elliptic curve factorization.

1.1.6.3 ElGamal encryption system

It is an asymmetric key encryption algorithm for public-key cryptography which is based on the Diffie-Hellman key agreement. In 1985 Taher Elgamal was the one to define it. ElGamal encryption is used in the unrestricted GNU Privacy Guard software, recent versions of PGP, and other cryptosystems. The Digital Signature Algorithm is a modified version of the ElGamal signature scheme, which should not be confused with ElGamal encryption.

1.1.7 Image Encryption

The primary thought in the image encryption is to transmit the image safely over the system so no unapproved client can ready to decode the image. The image information have uncommon properties, for example, mass limit, high severance and high association among the pixels that forces exceptional prerequisites on any encryption procedure [1]. The most well-known system of secure the advanced pictures is to scramble the computerized information such that unique message of the archives ought not to be identified. There are a few methodologies to accomplish this for instance steganography, packing, advanced watermarking and cryptography. Here the emphasis is on the encryption methods of advanced digital images focused around the chaos mapping. Fundamentally image encryption is the methodology of changing data utilizing a algorithm to make it ambiguous to anybody with the exception of those having exceptional learning, normally alluded to as a key and the changing data utilizing "encryption algorithm" into a structure that can't be deciphered without a key of decryption.

From the other point of view, decryption of image recovers the genuine data from the encrypted structure image. There are more than a few computerized image encryption frameworks to encode and decode the image information, and there is no single encryption calculation accessible that fulfills the distinctive image sorts. The encryption strategies focused around the chaos mapping gives the encoded advanced images to hold the multilevel encryption strategy furthermore diminishes the computational difficulty of the encryption process. A large portion of the algorithms particularly intended to scramble or encrypt computerized images are proposed in the mid-1990s. There are two significant assemblies of image encryption algorithms: (a) non-chaos selective methods and (b) Chaos-based selective or non-selective methods. The vast majority of these algorithms are intended for a particular image setup compacted or uncompressed, and some of them are even setup acquiescent.

There are systems that offer light encryption (degradation), although others compromise solid manifestation of encryption. A percentage of the algorithms are versatile and have different modes ranging from degradation to solid encryption [2]. The encryption methods focused around the chaos have distinctive sorts of uses in different zones, for illustrations ; the web correspondence, military, medicinal services, mapping and situating, picture informing applications on phones, interactive media frameworks, therapeutic imaging, Tele-pharmaceutical, protection and government archives and so forth. The advancement of image encryption procedure is moving towards a prospect of unlimited conceivable outcomes. On daily basis, new strategies for encryption methods are revealed [3].

1.2 Image Encryption Techniques

1.2.1 Classic Image encryption

Advanced Encryption Standard (AES) is a symmetric cryptosystem that proposed for content encryption by Rijmen and Daemen in 1999 [1] furthermore known as Rijndael algorithm, however a few scientists made functional use of this algorithm for image encryption likewise with a few changes in key generation and other requirements. Zeghid et al. [2] proposed an improved AES based algorithm by including a key stream generator (A5/1, W7) to AES to guarantee enhancing the encryption execution for image encryption process. An alternate algorithm proposed by Subramanyan et al. [3] focused around AES Key Expansion in which the encryption methodology is a bit astute XOR operation of a set of image pixels besides a 128 bit key that varies for each set of pixels. The keys to be utilized are produced freely at the sender and recipient side focused around AES Key extension transform thus the preliminary key is distant from everyone else imparted instead of offering the entire set of keys.

DES, a prevalent block cipher algorithm utilizes 64 bit key, which is an alternate printed cryptosystem that utilized for image encryption by Qian Gong-canister et al. In [4] another image encryption plan focused around DES consolidated with a chaotic map introduced to enhance the security and develop the key space. The results demonstrate that blend of word-based cryptosystems with different strategies or rolling out a few improvements, enhance the security and against anti attack capacity of those algorithms adequately.

1.2.2 Public Key Encryption

Most of application does not provide a facility of a secure channel to transfer the private key or desire to keep the decryption key in secret, so we need to utilize public key cryptography. In the first place public key was circulated by Diffie and Hellman in 1976 [5]. It was a key trade down to earth strategy for making an imparted secret key over a verified correspondence channel without utilizing a former imparted secret. The greater part of conventional public key cryptosystems intended to encode printed information. A few works have been distributed on public key image encryption, one is proposed by Shuihua et al. [6].

In this plan, the plain image isolated into blocks utilizing a certain network change and all pixels in each one block exchanged to DCT field. Public key, private key, encryption methodology and unscrambling procedure are characterized focused around change network of DCT coefficients. The results show that this system is vigorous in contradiction of JPEG lossy clamping and other general assaults. An alternate public key system focused around Chebyshev chaos map portrayed by K. Ganesan et al. [1] for colour images encryption and features progressively applications. In the first place they attempted to cryptanalysis the encryption focused around Chebyshev polynomial map and results demonstrate that it is not powerful on a few attacks, so they attempted to improve the security by utilizing a non-Xoring hash function to secure it against attack of picked plaintext. They do proficiency check and some testing for cryptanalysis, for example, key affectability, connection, mono bit, long run test and time examination for both image and video and determined from the result that their recommended cryptosystem is more secure and strong to any invader attack and the time investigation exhibits the effectiveness of encryption for 64x64 and 128x128 video encryption.

An image encryption strategy utilizing ECC is proposed by K. Gupta et al. [8] by transforming every pixel into the elliptic arc point to transform the plain image to encrypted image. They only suggested a framework and experiments done with a simple elliptic arc function with few points, so it is not an appropriate system, but as a innovative idea, results demonstrate the adequate encryption time in contrast with other public key techniques like RSA because of key size, and furthermore gives the key affectability yet needs to be upgraded as future works. Visual cryptography (VC) is a simple and safe technique proposed by Naor and Shamir [9] in 1994. In [10] A. Jaafar and A. Samsudin proposed another public key plan with straightforward and low processing by consolidation of VC and Boolean AND operation comes about a quick running time for encryption and decoding.

1.2.3 Compression and Encryption

Compression procedures help us to lessen the transmission data transfer capacity or storage space. These procedures can be actualized in both spatial and frequency domain. Also frequency domain procedures are further effectual and consuming collective and widespread transforms such as DCT, DFT and DWT. Data compression lessons can be classified into two types:

- LOSSY: Lossy methods compromise a definite loss for information in return with the high compression proportion. Usually lossy methods decline the superiority of the object so they are sought out for images, videos and audios for the reason of human observation. There Lossy coding method also moreover categorised into the following types:
  a. Predictive coding
  b. Transform coding
- LOSSLESS: On the other hand, some kinds of data could not accept any loss (e.g. medical images, database records, executable files and word processing files), otherwise the data will be corrupted. In such cases the lossless techniques plays

effective role. The Lossless coding technique also further classified into following categories:

a. Run length encoding

b. Huffman encoding

c. Arithmetic encoding

d. Entropy coding

e. Area coding

Ordinary cryptosystems identifies with the compressed multimedia. Encryption and compressed multimedia are typically extremely contradictory and an exchange off depends between them. Encrypting the interactive media content before pressure uproots a ton of repetition and this result in an exceptionally poor compression proportion. Then again, encrypting the information after compression demolishes the codec design, which are the bases for the decoders to crash.

As a final point, encryption is taken lightly for many applications to reserve approximate perceptual data [11]. B. Mohammed et al. [12] in their projected encryption-compression method initially enforced a FMT technique to compress the particular image and at that time AES-Based algorithm functionalized to encrypt the image. L. Vorwerk et al. [13] strained to syndicate encryption and wavelet compression. The methodology of encryption utilizes a symmetric key for encrypting image and wavelet filter, a public key cryptosystem is recommended to encrypt the symmetric key for secure key interchange.

A recent mixing arrangement of encryption and compression for images suggested by I. Masanori et al. [14] centred on Independent Component Analysis (ICA) and Discrete Cosine Transform (DCT). To attain a quick and secure image transmission they utilized DCT and a low pass filter for image compression and by rotating and making a mixture of the DCT blocks with an arbitrary image, the source image is encrypted. When this journey's end, the encrypted expected image is decrypted by taking out the protected images from the mixtures by spreading over ICA and lastly by utilizing rotation keys and IDCT the novel image is recreated.

Additional methodology to assimilate compression and encryption is deliberated in the system of C. Wu and J. Kuo [15]. They debated about benefits and drawbacks of discerning encryption and anticipated an encryption schemes which can transforms the entropy of simple message as an outcome to be a cipher message by spreading on Huffman coder and QM coder. Finally, concluded that this high security scheme can be applied to compression techniques such as MPEG and JPEG with acceptable computational speed.

1.2.4 Selective Encryption

A methodology that offered to abstain from encrypting the complete image is called selective encryption also acknowledged as partial encryption, soft encryption or perceptual encryption. The primary inspiration is to lessen the computation time for real-time applications that runtime performance is frequently serious deprived of compromising the security of the broadcast moreover. The primary objective is to divide the image content into two shares, public share and protected share. One significant feature in selective encryption is to decrease the protected part to least as it can be. Selective encryption generally accompanies compression. In frequency domain, low frequency coefficients convey most of the data of the image and high frequency coefficients convey the fine points [16].

In lossy compression techniques for example JPEG standard, an image changes to a frequency domain by DCT and at that point roughly high frequency coefficients are reproduced by zeros and new compressed image is recreated. Therefore only few low frequency coefficients can be encrypt relatively than all in frequency domain that also has many benefits [17]:

- It is less demanding to recognize the critical parts to be encrypted.
- It is less demanding to recognize parts of the information are not compressible.

In 1995, Maples et al. did the very first studies on selective multimedia encryption [18] by recommending Aegis mechanism focused around MPEG video transmission and DES cryptosystem to secure MPEG video sequences from unauthorized access.

This instrument cut off points the measure about information to be encrypted or decrypted toward utilizing feature layering to decrease the measure from claiming transmitted video images by encrypting intra I frames about an MPEG stream anyhow Agi and Gong [19] discovered that this and some other systems would not sufficient to touchy provisions furthermore might not make sufficiently secure for exactly sorts of video Also person can perceive example from movement patterns with the goal they attempted to move forward those security toward progress of I-frame frequency yet it reasons to increase in transfer speed utilization and also higher computational multifaceted nature. An elective approach will be to encrypting I-blocks on the whole frames rather that I-frames that enhances security. Droogen broeck likewise suggested two strategies to particular encryption from claiming both compacted also uncompressed images [20].

A selective encryption approach for uncompressed image is to encrypt 4 to 5 least significant bits because it is random like and plaintext attack on random like data is difficult. Another selective encryption method that mentioned in this paper is based on compressed JPEG images and encrypts a selected number of AC coefficients. Results on execution time on three different encryption algorithms (DES, 3-DES and IDEA) show that real-time processing is achievable without any difficulty. Droogen broeck [21] provided another technique for real-time applications that encrypts appended bits corresponding to a selected number of AC coefficients for each DCT block and he concluded that this scheme provides flexibility, multiplicity, spatial selectivity and format compliance.

A multilevel partial image encryption (MPIE) proposed in [22] that performs the encryption before compression. Encryption is performed on parts of low frequency coefficients that determined by Haar Wavelet, and DFT applied on the approximation coefficients and a permutation matrix as encryption key is used to permute the result of transformation and then compression is doing by Huffman coding. Regardless of limitations such as low rate of compression, time consuming of this algorithm and complexity, there are advantages such as security, flexibility to image transformations and compression techniques.

Another different approach in partial image encryption is; rather than encrypting the whole image, extract some special and secret features in an image and encrypt them. The main idea in this approach is to detect faces of input image and encrypt them for some military applications such as transmission of images with guilty and accused persons or members of security organizations. K. Hong and K. Jung [23] proposed a partial encryption method using the face region as a feature because a face has the semantic information and is the most important part in an image or video. They used Multi-Layer Perceptron's to detect face region and for more exact, Gaussian skin-color applied to discriminate between skin regions and non-skin regions.

Result shows that encryption time is less for DES when both AES and DES encryption algorithms are compared. Due to experiments, for video content encryption, fully encryption methods provide 2 or 3 frames in a second but their proposed method encrypts 25 to 30 frames per seconds. J. M. Rodrigues et al. [24] proposed a different scheme by for selective encryption Bin video offered also for face protection based on AES stream cipher for JPEG image sequences by performing three steps on DCT blocks. These steps are; construction of plain text, ciphering the plain text and substitution of the original Huffmans vector with the ciphered information. This scheme provides advantages such as constant bit rate, portability, and selective encryption of the region of interest and does not effects on entire JPEG compression rate, which makes it useful for a large range of applications with good information confidentiality results. S. Lian et al. [25] proposed a selective image encryption scheme based on JPEG2000 which is a widely used compression standard based on wavelet transform.

They lessened the encryption information ratio to short of what 20% by selecting huge bit-planes of wavelet coefficients in high and low frequency, so the encryption time proportion decrease by to short of what 12%. Their examinations demonstrate the security of their plan against savage power assault, select- plaintext assault or substitution assault and does not consequences for pressure proportion. An alternate late specific encryption focused around wavelet change in fractional wavelet area distributed by N.

Taneja et al. [26]. In this work, 3.125% of noteworthy image information chose by normalized information energy (NIE) and encoded these chose sub groups by Arnold cat map, a 2D chaotic function.

1.2.5 Digital Signature for Image Authentication

The demonstration of digital signature is like manually written on paper signature which assuming the principle part in validating reports and confirm the individuality. Subsequently, digital signature has numerous applications in data security. It is a system that offer verification, information reliability and on-revocation. Several years ago the first idea of digital signature was a plan focused on RSA [30] and today it is a standout amongst the most functional procedures.

The greater part of the digital signatures are focused around asymmetric cryptography. In these frameworks, the private key is utilized to make a digital signature that particularly proofs the underwriter who is holder of the private key and can be verified just with the relating public key. In 1998, C. Yung Lin and S. Chang put out an article which was one of the first studies on digital signature and its applications in images [31]. They proposed a vigorous digital signature focused around DCT coefficients in JPEG images. This produced digital signature is vigorous to trimming, strength changes, resizable and filter applicable.

Digital signature and watermarking are connected and both utilized for verification and confirmation however there is somewhat a contrast in their structure. Tao Chen et al. attempted to consolidate digital signature with watermarking [32]. The preference of this consolidation is to spare the obliged data transfer capacity for signature which is encoded to a different record. To attain this point, they install the signature record as a watermark, so their proposed plan can be able to verify the image, as well as can perform a copyright security. An alternate image verification plan [33] utilize the substance of an image wavelet change space to develop a structural digital signature. They demonstrated that this plan is forceful to content stabilizing controls and delicate to content evolving alterations.

H. Zang et al. proposed a plan focused around an arrangement of comprehensive synchronization Henon discrete-time chaotic system which utilizes as a pseudo-arbitrary number generator to build encryption and digital signature. [34]. The enormous key space as 10158, affectability to misperception of parameters and preliminary condition on account of applying chaos in encryption make this plan certain to be utilized as a part of secure correspondence.

1.3 Security Analysis of Encrypted Image

Security investigation is the specialty of discover the shortcoming of a cryptosystem and recovery of entire or a piece of a ciphered message (here we consider an image) or discovering the mystery key without knowing the decryption key or the algorithm. There are numerous methods to investigate, contingent upon what access the expert has to the plaintext, cipher content, or different parts of the cryptosystem. The following are probably the most widely recognized sorts of assaults on encrypted images:

1.3.1 Key Space Analysis

Attempt to discover the decryption key by checking all conceivable keys. The quantity of attempt to discover specifically denotes to key space of the cryptosystem become exponentially with aggregate key size. It implies that multiplying the key size for an algorithm does not just twice over the obliged number of operations, but instead squares them. An encryption algorithm with a 128 bit in key size characterizes a key space of 2128, which takes around 1021 years to check all the conceivable keys, with superior computers of these days. So a cryptosystem with key size of 128 bit computationally looks powerful against a beast energy assault.

1.3.2 Statistical Analysis

Original and encrypted image relationship can be determined by analysing data statistically. In this manner, image after encryption must be totally differentiate from the original. Because of Shannon hypothesis. It is conceivable to illuminate numerous sorts of images by statistical investigation. For a image there are a few approaches to figure out if the ciphered image releases any data about the first one or not.

1.3.3 Correlation Analysis

Two contiguous pixels in a plain image are intensively corresponded vertically and on a level plane. The most extreme estimation of relationship coefficient is 1 and the base is 0 considered as the property of an image, where a strong image that has been encrypted to measurable assault ought to have a connection coefficient estimation of 0.

1.3.4 Differential Analysis

The point of this examination is to focus the affectability of encryption algorithm to minor changes. On the off chance that an challenger can make a little change (e.g. one pixel) in the plain image to watch the results, this control ought to cause a noteworthy change in the image that has been encrypted and the challenger ought not to have the capacity to discover a compelling relationship between the original and the image that has been encrypted as for distribution and misperception, the distinct assault loses its productivity and get to be inadequate.

1.3.5 Key Sensitivity Analysis

Moreover of vast enough key space to oppose a cryptosystem at brute force attack, additionally a protected algorithm ought to be totally delicate to mystery key which implies that the encrypted image can't be decrypted by somewhat changes in mystery key.

Chapter 2

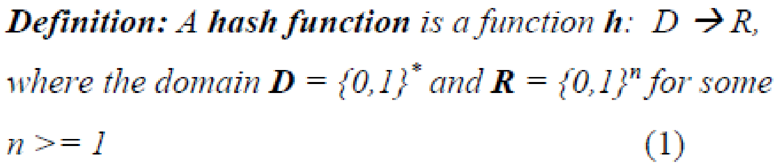
2 THEORY

2.1 Hash Functions

Hash Functions utilizes the concept of converting an uncertain amount of digital data to a preset amount of data, through making minor changes in input data features resulting in key changes in yield data. The values returned by a hash function are known as hash values, hash codes, hash sums, or purely hashes. One functional use is data structure called a hash table, generally utilized as programming tool on computers for quick information lookup. Hash functions speeds up the process of table or database lookup by identifying the replica records in a massive record. One scenario is locating alike stretches in DNA sequences. Moreover they have a very rational use in cryptography. A cryptographic hash function permits one to readily confirm that some input data matches a stored up hash value, yet makes it hard to recreate the information from the hash alone. This rule is utilized by the PGP algorithm for information corroboration and by numerous secret key checking frameworks.

Check digits, fingerprints, randomization functions, error-correcting codes, and ciphers are sometimes correlated to or mistaken as Hash functions. Despite the fact that these ideas have similarities at certain level, they are planned and improved in their own specific way. An example of such a scenario is the Hash Keeper database kept up by the American National Drug Intelligence Center, which is best depicted as a record fingerprints than of hash values. [39]

2.2 Cryptographic Hash Functions



Definition: *A* ***hash function*** *is a function* $\boldsymbol{h}$*:* $D \rightarrow R$*, where the domain* $\boldsymbol{D} = \{0,1\}^*$ *and* $\boldsymbol{R} = \{0,1\}^n$ *for some* $n >= 1$ (1)

A message digest $\{0,1\}^n$ is an algorithm H in which an arbitrary size message goes in as input {0,1}* and the result is generated in form of a fixed size output. This has a few common names such as a digital fingerprint, imprint, hash result, hash code, hash value, or simply hash. All these various names were in fact functions that were a crucial part in the contemporary cryptography and more realistic applications, for example, a digital signature, digital time stamp, message authentication code (or MAC), public key encryption, tamper detection of files and many more. It is sometimes called “Swiss army knife of cryptography” because of its adaptability to application. [42]

2.2.1 Two types of hash function:

2.2.1.1 Keyed and Un-keyed

Un-keyed hash function takes in a variable length message for input and gives back a fixed size hash digest, H: $\{0,1\}^* \rightarrow \{0,1\}^n$. Another name for un-keyed hash function is modification detection codes (MDCs). At the same time, keyed hash functions admits both variable and fixed length secret key as two different (pair of) inputs to the hash function design and generates a fixed length hash digest, HK: $\{0,1\}^k \times \{0,1\}^* \rightarrow \{0,1\}^n$. Message authentication codes (MACs) is the alternative name for keyed hash functions. [41]

Un-keyed are additionally categorized into

- OWHF (One Way Hash Functions)
- CRHF (Collision Resistant Hash Functions)
- UOWHF (Universal One way Hash Functions)

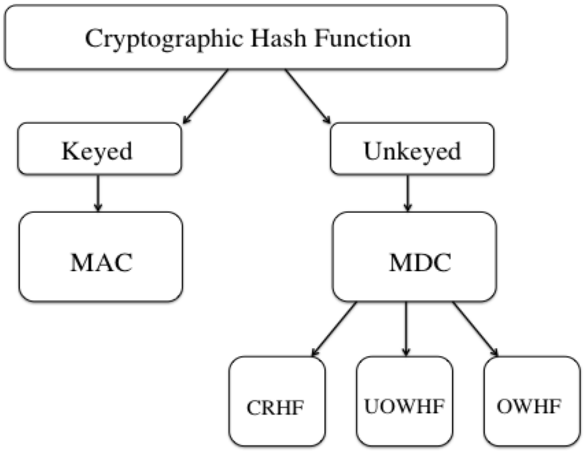


Figure 4: Types of Hash functions

2.2.2 Three main approaches of Security of Hash Functions

Nowadays security is an issue, so mostly applications are using hash functions.

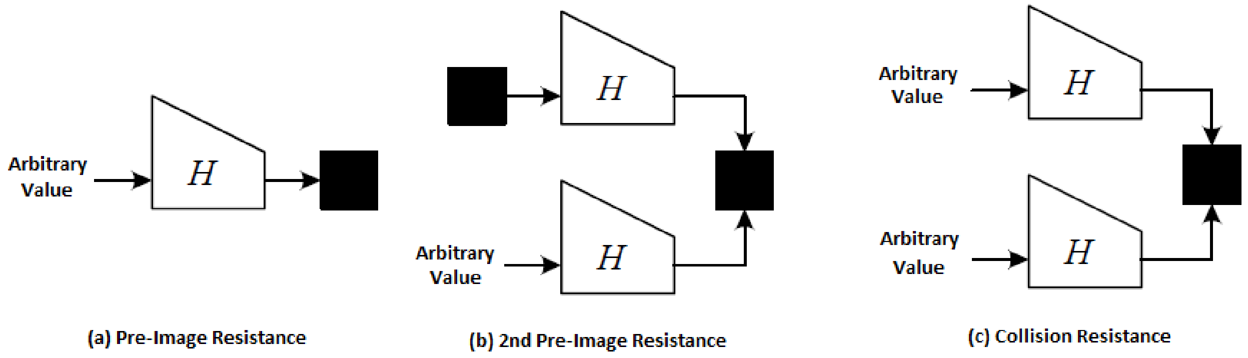


Figure 5: Security Techniques of Hash Functions

2.2.2.1 Pre-image Resistance

After applying hash function to a message it is impossible to obtain the original message back. For a pre-image resistant hash function H and a particular message M, there is a hash value H(M), it is impossible to retrieve the original message M, or indeed generate any message M'≠ M such that H(M') = H(M).Concisely,

$$\mathbf{Adv}_H^{\mathrm{pre}[m]}(A) = \Pr\left[M \overset{\$}{\leftarrow} \{0,1\}^m; Y \leftarrow H(M); M' \overset{\$}{\leftarrow} A(Y) : H(M') = Y\right]$$

A hash function is said to be pre-image resistant, when a brute force attack is used against it. Pre-image resistance is also sometimes called One-wayness. Generally, collision resistance does not guarantee pre-image resistance, but in it was shown that pre-image resistance can be implied by collision resistance if the hash function was sufficiently compressing (i.e. its domain is much larger than its range). [43]

2.2.2.2 2nd Pre-image Resistance

In a 2nd pre-image resistant hash function H, and a given message M, finding a distinct message M' such that M ≠ M' and both M and M' hash to the same value is impossible which makes it hard to be broken. Concisely,

$$\mathbf{Adv}_H^{\mathrm{sec}[m]}(A) = \Pr\left[M \xleftarrow{\$} \{0,1\}^m; M' \xleftarrow{\$} A(M) : M \neq M' \wedge H(M) = H(M')\right]$$

A hash function H deemed as 2nd pre-image resistant, when a brute force attack is used against it. 2nd pre-image resistance is also sometimes called Weak Collision Resistance. [43]

2.2.2.3 Collision Resistance

A hash collision takes place when two random messages hash to the same value. That is, for a collision resistant hash function H, it have to be impossible to find two distinct messages M and M' such that H(M) = H(M') while M ≠ M'. It is also applicable to the other relations in hash function (i.e. keyed hash functions, where members of the family are indexed by different keys). Concisely:

$$\mathbf{Adv}_H^{\mathrm{cr}}(A) = \Pr\left[(M, M') \xleftarrow{\$} A : M \neq M' \wedge H(M) = H(M')\right]$$

In order to have a well protected hash function, the most excellent attack is birthday attack. Collision resistance is sometimes called Strong Collision Resistance. Another term i-e multi-block collision to refer to 2 colliding messages, each consisting of at least 2 blocks. [43]

2.2.3 Security Services of Cryptographic Hash Function [40]

2.2.3.1 Achieving Integrity & Authentication

Reliability and authentication of the information is the key requirement in computer systems. It is very important to ensure the safe transfer of information from sender to receiver and this also requires a method to examine the legitimacy of that data. There are several ways to implement it. One way is symmetric encryption, but each method has its drawbacks such as speed, cost factor or optimization for sizes. Few of such strategies merge together Confidentiality and Authentication functions. Sometimes legitimacy of data is of more importance than security of message. To accomplish the goal of reliable and legitimate information sharing hash function is one of the trustworthy methods, with not making the use of symmetric encryption. It is also faster than block ciphers while executed inside software and its implementation.

2.2.3.2 Implementing Efficient Digital Signatures

The purpose of digital signature is to provide legitimacy and reliability to the sender and receiver. Also it provides them with a sense of protection for themselves as well. Hash function basically enhances and improves the schemes used to create the digital signature. The core procedure to apply this function is by signing the digest of the message and using it to create a signature. The receiver upon getting the message then computes the digest message by means of same hash function and cross examines it by verification using the algorithm. Such operations save time by reducing huge computing overheads.

2.2.3.3 Authenticate Users of Computer Systems

Hash functions sometimes are utilized to provide the confirmation of user login as passwords are kept as message digest. Every time client accesses it is registered and matched to maintain privacy and protection.

2.2.3.4 Digital Time Stamping

In order to maintain and provide the temporal authentication of digital media and text one way hash functions and digital signatures helps in protecting and implementing the way out and ease of maintaining the originality of content. Also time tamping is another way to protect intellectual property rights, ensuring strong auditing procedures and implementing true on-repudiation services.

2.2.3.5 Session Key Derivations

When there are consecutive sessions of data transfer or sharing, creating a series of session keys is the ideal solution. Starting from a master key K0, the first session key can be K1 = H(K0) and second session key can be K2 = H(K1) and so on. The key management scheme based on control vectors makes use of hash functions and Encryption functions for generating session keys. [40]

2.2.3.6 Other Applications

Hash Functions has several other applications as well in terms of privacy, authentication and reliability. It also has the capability to index data in hash tables, for fingerprinting, to detect duplicate data or uniquely identify files, and as checksums to detect accidental data corruption and for generating random numbers also.

Hash functions have large diversity of applications which cannot be characterized under any precise sub branch of cryptography. Whenever resourceful information is needed Hash function pops to mind due to its diverse applicability.

2.2.4 Iterative Structure of Hash Functions

2.2.4.1 MerkleDamgard Iterated Hash Design [41]

Merkle-Damgård construction was proposed in 1979. The Figure below shows its structure.

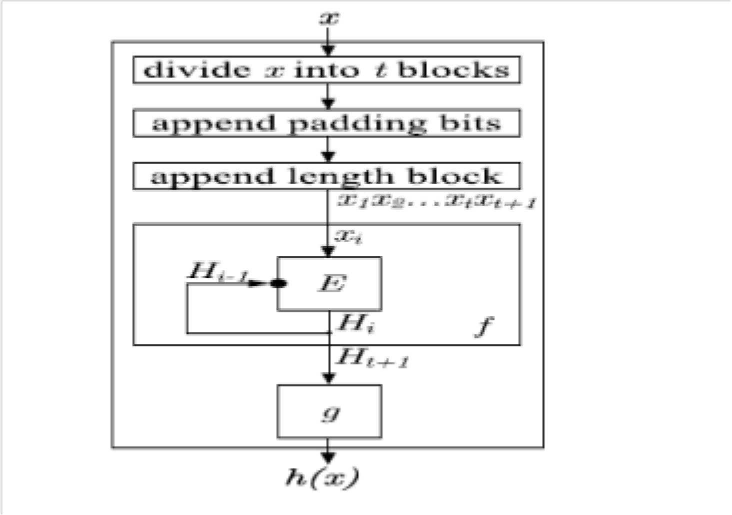


Figure 6: Detailed View of Merkle-Damgård Structure

The algorithm steps of Merkle-Damgård structure are following:

a. Break the input x into blocks x_1, x_2…..x_t.

b. Pad the last block x_t with 0-bits if necessary to obtain the multiple length of r.

c. Create the length block x_{t+1} with bit length r to hold the right justified binary representation of overall bit-length of x (MD strengthen).

d. Inputting x_1, x_2…..x_t to the compression function (iterated processing) to produce an intermediate value of H_i.

e. Hi serves as feedback value to f and is processed with xi+1 in the next iteration. This implies the need of an initial value (IV) H0 for the first iteration that is often provided pre-defined with bit- length r.

f. After processing all the input blocks, then, function g transforms the preliminary result H_{t+1} of bit-length r to the final hash-value with desired bit-length. Function g is often the identity mapping.

2.2.4.2 *Wide Pipe Iterated Hash Design [41]*

Stefan lucks introduced the wide pipe iterated hash design evolved from Merkle-Damgård. The purpose was to enhance the structural flaws in the Merkle-Damgård.

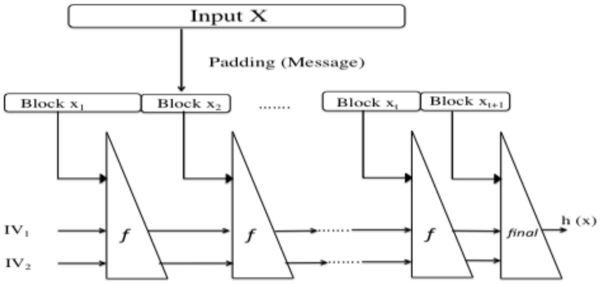


Figure 7: The wide pipe hash Structure

The process is similar to Merkle-Damgård algorithm but with an improvement by having a larger internal state size, which results in a smaller final hash digest. The internal state size of bit length is now larger than the digest. Also, the final compression function compresses the internal state length to output a hash digest which is simply half of internal state length.

2.2.4.3 *Hash Iterated Framework (HAIFA) [40]*

HAIFA was proposed in 2006. Its structure is used to overcome many of the drawbacks detected in MerkleDamgard Construction. The core design of HAIFA is to initiate with the number of bits that were hashed so far and a salt value into the compression functions. In HAIFA chaining value H_i is computed as

$$H_i = f(H_{i-1}, M_i, \#bits, salt)$$

Where #bits is number of bits hashed so far and salt is a salt value.

2.2.4.4 *Fast Wide Pipe (FWP) Design [41]*

The fast wide pipe structure is twice as fast as the wide pipe construction. Figure shows the fast wide pipe structure.

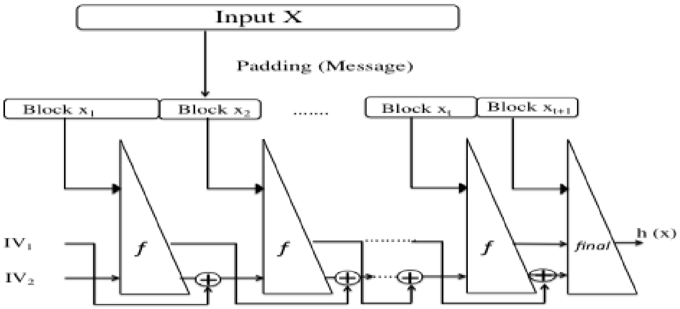


Figure 8: The fast wide pipe hash Structure

As the Figure shows, the input (IVs) for each compression function is divided into halves. The first half is inputted in the compression function and the other half is XORed with the output for the same compression function. The feed-forward process makes the overall design faster. Hence, faster process is obtained. The final output of the hash digest can be truncated to the desired digest length using the final compression function.

2.2.4.5 Sponge Construction [41]

Sponge construction is basically an iterative construction devised to replace Merkle-Damgård construction. This technique actually maps an arbitrary length input to an arbitrary length output. By the use of fixed-length transformation/permutation f that operates on a preset number of b = r + c bits. Where r is the bit rate and c is the capacity. First, the input is padded with padding algorithm and cut into blocks of r bits. Then, the b bits of the state are initialized to zero. The sponge construction operates in two phases:

a. Absorbing phase: The r-bit message blocks are XORed with the first r bits of the state of the function F. After processing all the message blocks, the squeezing phase starts.

b. Squeezing phase: The first r bits of the state are returned as output blocks of the function F. lastly, the number of output blocks is chosen by the user.

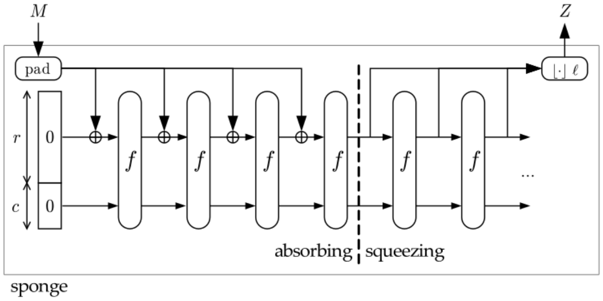


Figure 9: The Sponge Structure

After quite a few experiments its security robustness is established.

2.2.4.6 Other Constructions [40]

In addition to the above listed Iterative Hash constructions, few more like Enveloped

MerkleDamgard, RMC construction and ROX construction have been suggested in literature. Cascaded Constructions have also been discussed in the literature to build large hash values by concatenating concatenate several smaller hashes. For example, given two hash functions H1 and H2, the concatenation H1(M) || H2(M) can be used to generate large hash value for message M. In this construction, H1 and H2 can either be two completely different hash functions or two slightly different instances of the same hash function. But Joux using multi-collisions proved that If H1 and H2 are good iterated hash functions with no attack better than the generic birthday paradox attack, then the large hash function H1|| H2 obtained by concatenating H1 and H2 is not really more secure that H1 or H2 by itself.

2.2.5 Security Properties of Hash Functions

2.2.5.1 Basic Security Properties

Fundamental concept of security of Hash functions is centred around pre-image resistance, second-pre-image resistance and collision resistance (as defined in Section 2.2). Pre-image is a one way journey, but 2nd pre-image and collision resistance are also called weak and strong resistance respectively. A function that is collision resistant is 2nd pre-image resistant as well. This rule doesn't apply on 2nd pre-image and pre-image as they have opposite of one another. While implementing collision resistance is the strongest property of all three, hardest to satisfy and easiest to breach, and breaking it is the goal of most attacks on hash functions.

2.2.5.2 Avalanche Criterion and Completeness

For hash function it is considered necessary to have unlike outputs for unlike inputs irrespective of dissimilarity in inputs. This makes up the following two properties of hash functions i.e. Completeness and Avalanche effect. Strong Avalanche effect implies that a minute change in the input will have a major impact on the message digest. Whereas Completeness implies that a single bit change and each bit individually has its cause and effect on the output bits. Now the strict Avalanche criterion is in fact the combination of the above two mention methods i-e with a change in one bit of input the outcome has modification in every bit of the output (message digest) with a probability of ½. If these criterions are not satisfied then the probability of successful attack on the hash functions increases considerably.

2.2.5.3 Certificational Properties and weaknesses

Besides fundamental properties, there exist some certificational properties too. These certificational properties are proposed to be near collision resistance, partial pre-image resistance, free start collision resistance, pseudo collision resistance, semi Free start collision. Certificational weaknesses are in fact the absence of these properties. As much as these properties are part of hash functions but it is not mandatory to be part of it. Certificational weaknesses cause uncertainty in the design principles which could lead to full collision in specific situations.

2.2.6 Methods of attack on Hash Functions

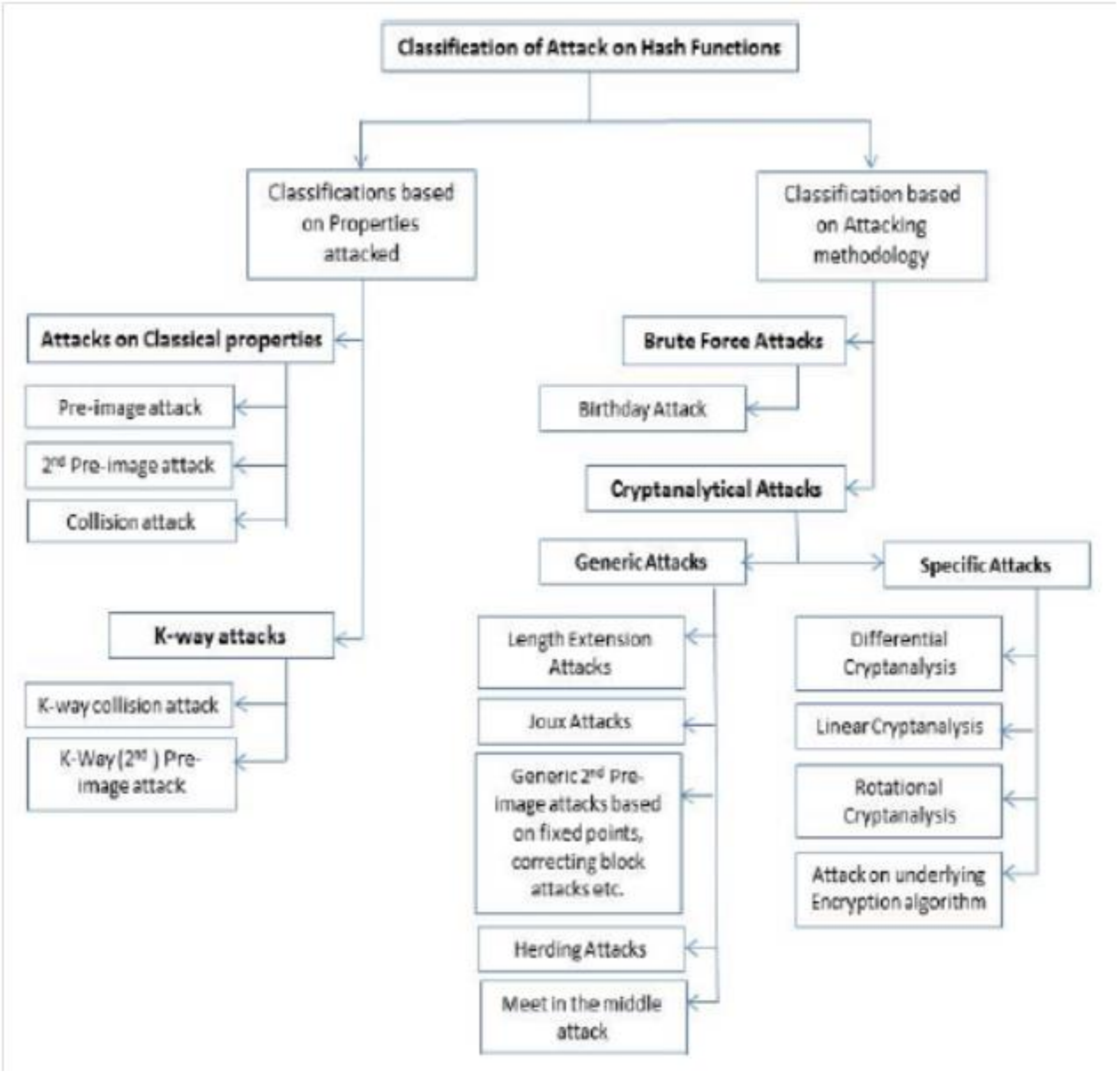

Classification of Attack on Hash Functions
Classifications based on Properties attacked
Classification based on Attacking methodology
Attacks on Classical properties
Pre-image attack
2nd Pre-image attack
Collision attack
K-way attacks
K-way collision attack
K-Way (2nd) Pre-image attack
Brute Force Attacks
Birthday Attack
Cryptanalytical Attacks
Generic Attacks
Specific Attacks
Length Extension Attacks
Joux Attacks
Generic 2nd Pre-image attacks based on fixed points, correcting block attacks etc.
Herding Attacks
Meet in the middle attack
Differential Cryptanalysis
Linear Cryptanalysis
Rotational Cryptanalysis
Attack on underlying Encryption algorithm

Figure 10: Classification of Attack on Hash Functions

2.2.6.1 Brute Force Attack

Brute force attack can happen on any hash function whatever their working and structure is. This attack can do an exhaustive search to hit upon the secret key of encryption scheme. The security of any hash function lies in its output bit size. In order to attempt an attack on a hash code of length n, the height of resistance needed depends on the type of encryption on it.

Pre-image attack: Effort required for brute force attack = 2^n. In this attack, for a given n-bit digest h of the hash function H( ), the attacker evaluates H( ) with every possible input message M until the attacker obtains the value h.[40]

2nd Pre-image attack: Effort required for brute force attack = 2^n.In this attack, for a given message M and the hash function H( ), the attacker tries H( ) with every possible input message M' ≠ M until the attacker obtains the value H(M).[40]

Collision attack: Effort required for brute force attack = $2^n/2$. In this attack, for a given hash function H, the attacker tries to find two messages M and M' such that M ≠ M' and H(M) = H(M'). On average the opponent would have to try 2^n / 2 (= 2^{n-1}) messages to find one that matches the hash code of the intercepted message However a chosen plain text attack (based on Birthday Paradox) is possible and in that case the effort required for collision in a Hash function is $2^n/2$ in place of 2^{n-1} [45]. It is also referred as Birthday attack.

K-Way Collision attack for K >=2:

Find K different messages M^1 such that $H(M_1) = \ldots = H(M_K)$. [44]

K-Way (2nd) pre-image attack for K>=1:

Given Y (or M with H(M) = Y), find K different messages M^i, with $H(M^i) = Y$ and $M^i \neq M$.[44]

2.2.6.2 Cryptanalytical Attack

Cryptanalysis of Hash functions focuses on the underlying structure of hash function and/or on the algorithm of Compression Function. Due to fixed size of the hash values compared to much larger size of the messages, collisions must exist in hash functions.

However, for the security of the hash function, they must be computationally infeasible to find. Collisions in hash functions are much easier to find than pre-images or 2nd pre-images.

Informally, a hash function is said to be "broken" when a reduced number of evaluations of the hash function compared to the brute force attack complexities and the strengths estimated by the designer of the hash function are used to violate at least one of its properties immaterial of the computational feasibility of that effort. For example, assume that it requires 290 evaluations of the hash function to find a collision for a 256-bit hash function. Though it is impractical to generate this amount of computational power today, the hash function is said to be broken as this factor is less than the 2128 evaluations of the hash function required by the Birthday attack. It should be noted that hash functions are easier to attack practically than encryption schemes because the attacker does not need to assume any secrets and the maximum computational effort required to attack the hash function is only upper bounded by the attacker's resources not users gullibility. This is not the case with block ciphers where the maximum practical count of executions of the block algorithm is limited by how much computational effort the attacker can get the user to do.

Collision finding algorithm and attacks may be classified as single block attacks or multi block attacks depending on whether that attack uses single block (i.e. one compression function) or more than one block (i.e. more than one iteration of compression function) for finding collision or pre-images. Gauravaram in his Ph.D. thesis has further classified Cryptanalytical attacks on hash function in two categories i.e. Generic and Specific attacks.

2.2.6.2.1 Generic Attacks

The attacks that work on a general hash function construction are called generic attacks. For example, attacks on the Merkle-Damgard construction that work on all hash functions designed using Merkle-Damgard construction are the generic attacks. Generic attacks are applicable even if we replace the underlying compression function by some abstract oracle. Length extension attacks, Joux multi-collision attacks, Generic 2nd pre-image attacks like the one based on Fixed points, correcting block attack, Herding Attacks and Meet in the Middle attacks are example of Generic cryptanalysis attacks.

- Length Extension Attacks
- Joux Multi-collision Attacks
- Multi (2nd) pre-image Attacks based on Joux Technique
- Generic 2nd pre-image Attacks
    - Correcting block attack
    - Fixed Point Attacks
- Herding Attacks
- Meet in the Middle Attack

2.2.6.2.2 Specific Attacks

The attacks that work on specific hash function or the algorithm of its compression function are called specific attacks. For example, collision attacks on the specific hash functions MD4,MD5, SHA-0 and SHA-1. Attacks using differential cryptanalysis, linear cryptanalysis, rotational cryptanalysis &attack on the underlying encryption algorithms are type of specific cryptanalysis attacks. The most successful of these are the attacks based on differential cryptanalysis

- Differential Cryptanalysis
- Linear Cryptanalysis
- Rotational Cryptanalysis
- Attacks on underlying Encryption Algorithm

2.2.7 Type of Hash functions based on design of underlying Compression Function

2.2.7.1 Hash Functions based on Block and Stream Cipher as Compression functions [43]

Hash functions based on block ciphers is very common and well-known methodology. There are several methods which can be used to create them. One way is to use compression function as a block-cipher. There a two inputs for this function i-e a message block and a key.

These 64 constructions are sometimes called PGV constructions after the authors' initials who used an attack-based 14 analysis approach to study the security of these constructions. It was then reported that 12 out of the 64 PGV constructions are collision resistant, but later Black et al. [25] showed (using proof-based approach this time) that another 8 PGV constructions are also collision resistant if they were properly iterated, even if their underlying compression functions are not collision resistant. The most widely adopted construction of these 20 PGV construction is the one attributed to Davies and Meyer [78]: $yi = f(h_{i-1},M_i) \oplus y_{i-1}$, where y_{i-1} and Mi are the input of the compression function f, and yi is its output. Another popular PGV construction is the Matyas-Meyer-Oseas construction [25], which is the opposite of the Davies-Meyer one. In Matyas-Meyer-Oseas, the output of the compression function y_i is further XORed with the message block input Mi (rather than the chaining variable y_{i-1} in Davies-Meyer). Further analyses and proofs of the collision resistance and pre-image resistance of these PGV constructions in the ideal cipher model can be found in [50] and [107].

Although PGV functions are provably secure, they are inefficient because the key (which represents the message block input of the compression function) is changed with every compression function call, and this is undesirable with block-ciphers since changing the key rapidly requires huge amount of computation (due to key setup). Thus, another approach is to use fixed-key block-cipher based compression functions [24, 100, 105, 106]. In this approach, a small non-empty set of keys are fixed and used for the block-cipher (when called by the compression function), while wrapping the block-cipher with other arbitrary functions to process the other compression function's input that was previously used as a key (which is now fixed). However, Black et al. [24] proved that such construction, making a single call to the fixed-key block-cipher, although efficient, cannot be collision resistant.

An inherent problem with designing hash functions based on block-ciphers is that block ciphers usually have small block size (e.g. 128 bit) which is insufficient to maintain an acceptable hash function security15, unless the result of the hash function can be expanded, which proved to be even more difficult. A particularly interesting solution to this dilemma is designing double block length (DBL) compression functions where the compression function outputs double the size of the underlying block-cipher [81, 62]. Clearly, however, DBL based hash functions still scarify some efficiency. Although the stream-cipher based approach is less popular than the block-cipher based approach, in the recent SHA-3 competition, some of the successful second round candidates were based on stream-ciphers (e.g. Cube Hash [19]). The main differences between block-cipher-based and stream-cipher-based hash functions are the size of the block and the number of rounds. In block-cipher-based, the message blocks are usually large, and iterated a small number of rounds, while in stream-cipher-based, the block size is small, with more rounds. Thus, in block cipher- based, a good compression function is necessary but in stream-cipher-based, even a weak compression function may provide sufficient security.

2.2.7.2 Hash functions based on Modular Arithmetic [40]

Compression function can also be designed using modular arithmetic. This allows the reuse of existing implementations of modular arithmetic such as in asymmetric cryptosystems. The idea of cryptosystems based on modular arithmetic is to reduce the security of a system to the difficulty of solving the problems in number theory. Two important hard problems in number theory which can act as a base for generating cryptosystems are factorisation and Discrete logarithm. Rompay in [41] has referred to design of two variants of MASH hash functions based on modular arithmetic. The advantage of such hash functions is that the level of security can be easily enhanced by choosing Modulus M of appropriate length but hash functions based on modular arithmetic are very slow, even slower than block cipher based hash functions. Also many such constructions have been broken in past.

2.2.7.3 Few Other approaches

Occasionally, attempts were made to adopt less common approaches when designing hash functions, most of which haven not attracted much interest. In this section, we discuss two such approaches: chaos-based and cellular automata based hash functions.

Chaos-based Hash Functions. Chaos theory is the mathematical representation of dynamic systems. These systems possess many desirable properties that suit the requirements of hash functions. For example, chaotic systems are very sensitive to changes in their initial values, potentially fulfilling the desirable hash function property requiring the output of the hash function to be highly sensitive to changes in its input; this phenomena is called the avalanche effect (also called butterfly effect in the chaos theory literature). Moreover, chaotic systems are one way functions and unpredictable. Hash functions based on chaos theory use chaotic maps, which are functions that exhibit particular chaotic behaviours. Unfortunately, most chaos-based hash functions suffer from poor efficiency due to their inherent complex structure, which makes them unattractive as a practical approach for building hash functions.

Cellular automata based hash function: Cellular Automata (CA) are discrete time models consisting of collections of cells organised in a grid, and each cell has a current state. The states of the cells evolve over time depending on their current states and the states of the neighbouring cells.

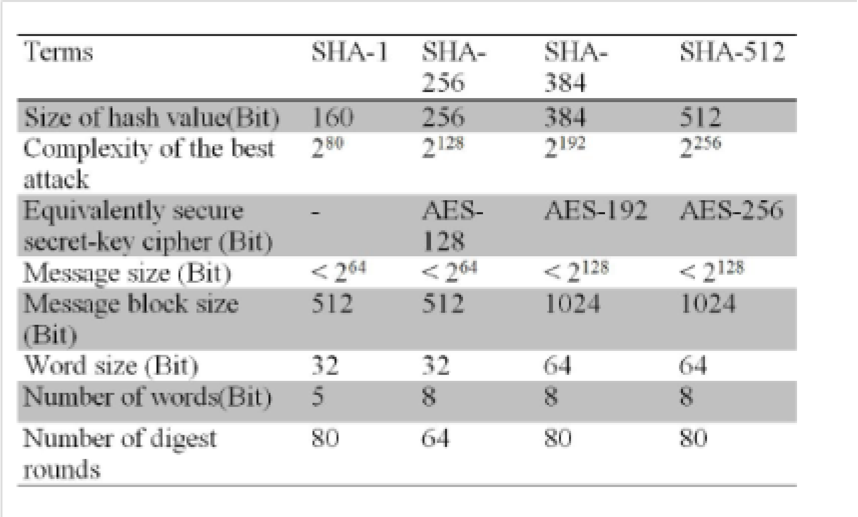
Chapter 3

3 LITERATURE SURVEY

3.1 A Novel Image Encryption Algorithm Based on Hash Function

This paper presents a fast image encryption algorithm. By means of SHA-2 it is intended to create an encryption mask such that size is reduced to half the original image size. Image encryption is performed by applying this mask over the digital image. This techniques has high security level, high speed, and high sensitivity. Simulation results showed how easy is the implementation of this algorithm.

3.1.1 Hash functions



Terms	SHA-1	SHA-256	SHA-384	SHA-512
Size of hash value(Bit)	160	256	384	512
Complexity of the best attack	2^{80}	2^{128}	2^{192}	2^{256}
Equivalently secure secret-key cipher (Bit)	-	AES-128	AES-192	AES-256
Message size (Bit)	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
Message block size (Bit)	512	512	1024	1024
Word size (Bit)	32	32	64	64
Number of words(Bit)	5	8	8	8
Number of digest rounds	80	64	80	80

Figure 11: Functional characteristics of four investigated hash functions

Functional characteristics of four hash functions are evaluated in above stated Figure 11. There are two main operational modes of SHA-2: preprocessing and hash computation. Preprocessing comprises of padding the input message, decomposing the padded data into several block of size m-bits, and locating and placing the suitable initial values used for hash generation. The second mode that is hash computation utilizes the padded data alongside of functions, constants, and word logical and algebraic operations. The purpose is to iteratively generate series of hash values. This transformation is applied a given number of times until created hash value is equal to the message digest [48]. These two properties ensure high security, input sensitivity and increased performance for proposed algorithm.

3.1.2 The proposed algorithm

3.1.2.1 Architecture of substitution - diffusion type hash-based image cryptosystem

The architecture of substitution-diffusion type SHA-2(512) based image cryptosystem is shown in Fig. 12. It comprises of four stages. First two stages have both substitution and diffusion process. The last two stages have only diffusion process.

In the substitution process, one fourth of the image pixels are replaced according to the S-box shown in Fig. 13. The purpose of the diffusion process is to change the pixel value in s sequence such that a minute change in one pixel is spread out to a lot of pixels, optimistically the entire image. To decorrelate the connection between neighbouring pixels, the substitution process is carried out times, where is usually larger than 1. Diffusion process follows the substitution process. The entire substitution-diffusion process reiterates several times so as to attain a acceptable level of security. In this paper, with no loss generalization, assuming source image is purely an image. For a big image, we can split it into blocks of size then the input image is divided into equal four sub images(namely Se.1, Se.2, Se.3 and Se.4 images). With the intention of finer comprehension of the formation, initially equivalent description of the cryptosystem is explained and then the cores of each stage is initiated.

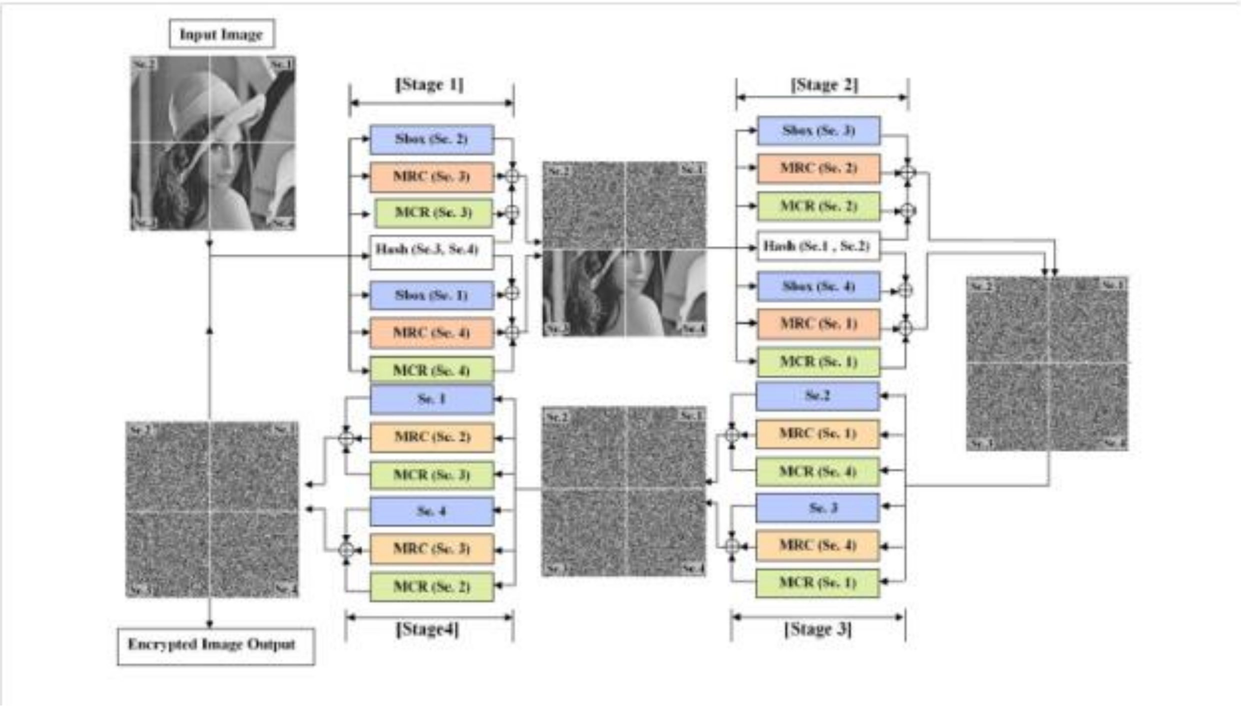
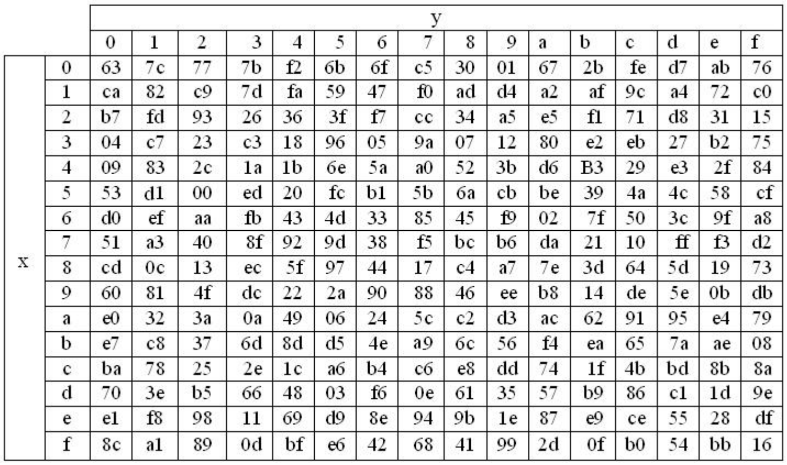


Figure 12: Architecture of substitution-diffusion type SHA-2 (512) based image cryptosystem



		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	B3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Figure 13: S-box: Substitution values for byte

1. Equivalent description of the cryptosystem:

   Equivalent description of Fig.11 structure is as follows:

- Substitute the sub-image Se.x according to S-box of AES (Sbox(Se.x)): Substitute a quarter of the image pixels according to the S-box of AES.
- The mean of columns of each row in Se.x sub-image (MCR(Se.x)): For Se.x sub-image, XOR all the gray level values of row i, where x = 1, 2, 3, 4 and i = 1, 2, · · · , 128. As a result, a matrix of size 128 × 1 is obtained. For a mask, concatenate the matrix horizontally 128 times.
- The mean of rows of each column in Se.x sub-image (MRC(Se.x)): For Se.x sub-image, XOR all the gray level values of column i, where x = 1, 2, 3, 4 and i = 1, 2, · · · , 128. As a result, a matrix of size 1 × 128 is obtained. For a 128 × 128 mask, concatenate the matrix vertically 128 times.
- Sub-images Hashing (Hash(Se.x, Se.y)): Sub-images Se.x and Se.y forms a matrix of size 128 × 256 called M. Each row of M is divided into four sub arrays with size of 124 bytes. In addition, four 8-bit keys will be appended to each of these sub arrays.

  Four sub arrays (SA derived from row i in M and its appending keys are shown as below:

  SA1 = [M(i, 1 : 124),Key(1),Key(2),Key(3),Key(4)]

  SA2 = [M(i, 51 : 174),Key(5),Key(4),Key(7),Key(8)]

  SA3 = [M(i, 100 : 223), Key(9),Key(10),Key(11),Key(12)]

  SA4 = [M(i, 132 : 255),Key(13),Key(14),Key(15),Key(16)]

The output of hash function for each sub array is 64 bytes. Hence the hash function's output for the row i will be 256 bytes. This way hash function for matrix M with 128 rows will create the mask of size 128 × 256.

- Substitute the sub-image Se.x according to Inverse Sbox of AES (InvSbox (Se.x)): Substitute quarter of image's pixels according to the Inverse S-box of AES.

2. Introducing the cores of each stage:

   The substitution process is comprehended exclusively by permutation of all the pixels by Sbox(Se.x), where x = 1, 2, 3, 4. During the diffusion process, the masks of MCR(Se.x), MRC(Se.x) and Hash(Se.x,Se.y) are mapped to sub-image Se.x. Fig.1 shows that each stage has 2 core processes that run simultaneously. Each core in stages 1 and 2 consists of both substitution and diffusion process whereas each core in stage 3 and 4 includes only diffusion process. Hash functions are used in both stages in the above structure for cryptosystem. Considering the sensitivity of the hash function to the input value, it can be concluded that our proposed structure has high sensitivity in encryption process.

3.1.2.2 Encryption

In the proposed cryptosystem structure, the width and length of the image have to be multiple of 256, or else they should be adjusted. Herein a 128-bit key is used. First, the image is segmented into sub-images of 128 × 128 pixels. Then the upper half part of the image is encrypted using the information of the lower half part. Similarly the lower half part will be encrypted by using the information of the upper half part.

The detailed encryption algorithm is described as follows:

Step 1. First choose secret keys key(1), key(2), ..., key(16) randomly for the input of hash function as described in Section 3.1.1. After that substitute sub-images Se.1old and Se.2old according to Sbox($Se.1_{old}$) and Sbox($Se.2_{old}$). Next calculate MCR($Se.4_{old}$), MCR($Se.3_{old}$), MRC($Se.4_{old}$), MRC($Se.3_{old}$) and Hash($Se.3_{old}$, $Se.4_{old}$)). The new sub-images Se.1new and Se.2new are obtained according to:

$$Se.1_{new} = Sbox(Se.1_{old}) \oplus MCR(Se.4_{old}) \oplus MRC(Se.4_{old}) \oplus Hash(Se.3_{old}, Se.4_{old})$$

$$Se.2_{new} = Sbox(Se.2_{old}) \oplus MCR(Se.3_{old}) \oplus MRC(Se.3_{old}) \oplus Hash(Se.3_{old}, Se.4_{old})$$

Step 2. Substitute sub-images $Se.3_{old}$ and $Se.4_{old}$ according to Sbox($Se.3_{old}$) and Sbox($Se.4_{old}$). Then calculate MCR(Se.1old), MCR($Se.2_{old}$), MRC($Se.1_{old}$), MRC($Se.2_{old}$) and Hash($Se.1_{old}$, $Se.2_{old}$)). The new sub images Se.3new and Se.4new are obtained according to:

$$Se.3_{new} = Sbox(Se.3_{old}) \oplus MCR(Se.2_{old}) \oplus (3)\ MRC(Se.2_{old}) \oplus Hash(Se.1_{old}, Se.2_{old})$$

$$Se.4_{new} = Sbox(Se.4_{old}) \oplus MCR(Se.1_{old}) \oplus (4)\ MRC(Se.1_{old}) \oplus Hash(Se.1_{old}, Se.1_{old})$$

Step 3. Calculate MCR($Se.1_{old}$), MCR($Se.4_{old}$), MRC($Se.1_{old}$) and MRC($Se.4_{old}$). The new sub-images $Se.2_{new}$ and $Se.3_{new}$ are obtained according to:

$$Se.2_{new} = Se.2_{old} \oplus MCR(Se.4_{old}) \oplus MRC(Se.1_{old})$$

$$Se.3_{new} = Se.3_{old} \oplus MCR(Se.1_{old}) \oplus MRC(Se.4_{old})$$

Step 4. Calculate MCR($Se.2_{old}$), MCR($Se.3_{old}$), MRC($Se.2_{old}$) and MRC($Se.3_{old}$). The new sub-images $Se.1_{new}$ and $Se.4_{new}$ are obtained according to:

$$Se.1_{new} = Se.1_{old} \oplus MCR(Se.3_{old}) \oplus MRC(Se.2_{old})\ (7)$$

$$Se.4_{new} = Se.4_{old} \oplus MCR(Se.2_{old}) \oplus MRC(Se.3_{old})\ (8)$$

Step 5. Repeat the above steps R times with fixed keys to satisfy the security requirements.

3.1.2.3 Decryption

The decryption scheme of our cryptosystem is shown in Fig. 14. The decryption procedure is similar to that of the encryption but the orders are reversed and InvSbox is used instead of Sbox. The detailed decryption algorithm is described as follows:

Step 1 and Step 2. These steps are the same as step 4 and step 3 in encryption process respectively.

Step 3. According to the approach described in Section 3.1.1 this step uses secret keys key(1), key(2), ..., key(16) for hash function input. Calculate MCR($Se.1_{old}$), MCR($Se.2_{old}$), MRC($Se.1_{old}$), MRC($Se.1_{old}$) and Hash($Se.1_{old}$, $Se.2_{old}$). By using InvSbox new sub-images

$Se.3_{new}$ and $Se.4_{ne}$w are obtained according to:

$Se.3_{new} = InvSbox(Se.3_{old} \oplus MCR(Se.2_{old}) \oplus MRC(Se.2_{old}) \oplus Hash(Se.1_{old}, Se.2_{old}))$

$Se.4_{new} = InvSbox(Se.4_{old} \oplus MCR(Se.1_{old}) \oplus MRC(Se.1_{old}) \oplus Hash(Se.1_{old}, Se.2_{old}))$

Step 4. Calculate MCR($Se.4_{old}$), MCR($Se.3_{old}$), MRC($Se.4_{old}$), MRC($Se.3_{old}$) and Hash($Se.3_{old}$, $Se.4_{old}$). The new sub-images $Se.1_{new}$ and $Se.2_{new}$ are obtained according to:

$Se.1_{new} = InvSbox(Se.1_{old} \oplus MCR(Se.4_{old}) \oplus MRC(Se.4_{old}) \oplus Hash(Se.3_{old}, Se.4_{old}))$

$Se.2_{new} = InvSbox(Se.2_{old} \oplus MCR(Se.3_{old}) \oplus MRC(Se.3_{old}) \oplus Hash(Se.3_{old}, Se.4_{old}))$

Step 5. Repeat the above steps R times with fixed keys to obtain decrypted image output.

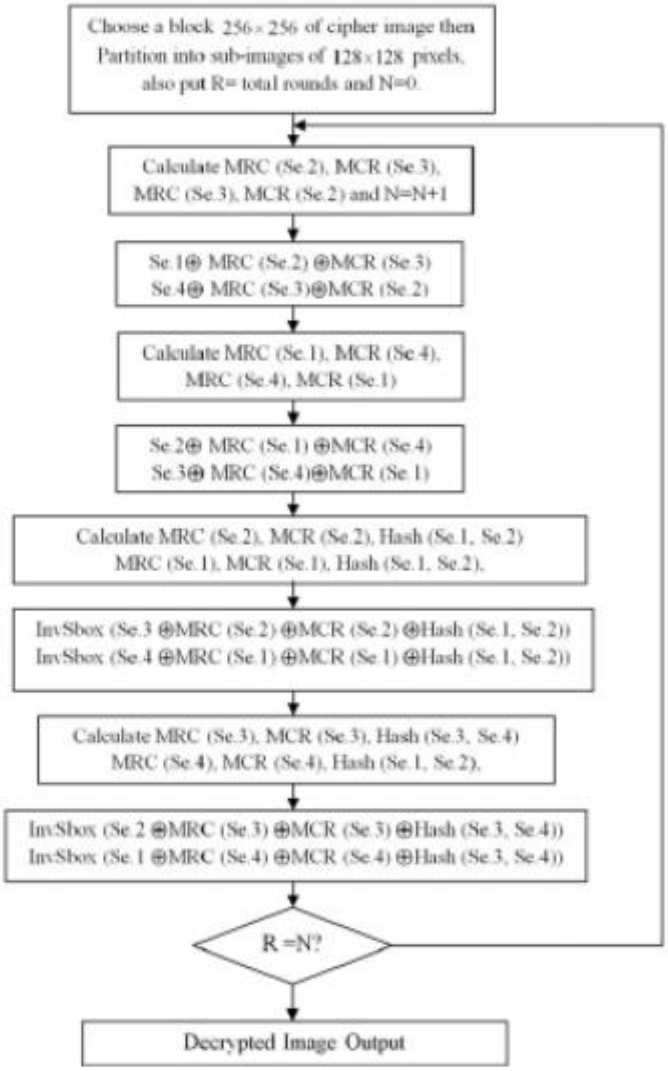

Choose a block 256×256 of cipher image then Partition into sub-images of 128×128 pixels, also put R= total rounds and N=0.
Calculate MRC (Se.2), MCR (Se.3), MRC (Se.3), MCR (Se.2) and N=N+1
Se.1⊕ MRC (Se.2) ⊕MCR (Se.3)
Se.4⊕ MRC (Se.3)⊕MCR (Se.2)
Calculate MRC (Se.1), MCR (Se.4), MRC (Se.4), MCR (Se.1)
Se.2⊕ MRC (Se.1) ⊕MCR (Se.4)
Se.3⊕ MRC (Se.4)⊕MCR (Se.1)
Calculate MRC (Se.2), MCR (Se.2), Hash (Se.1, Se.2) MRC (Se.1), MCR (Se.1), Hash (Se.1, Se.2).
InvSbox (Se.3 ⊕MRC (Se.2) ⊕MCR (Se.2) ⊕Hash (Se.1, Se.2))
InvSbox (Se.4 ⊕MRC (Se.1) ⊕MCR (Se.1) ⊕Hash (Se.1, Se.2))
Calculate MRC (Se.3), MCR (Se.3), Hash (Se.3, Se.4) MRC (Se.4), MCR (Se.4), Hash (Se.1, Se.2).
InvSbox (Se.2 ⊕MRC (Se.3) ⊕MCR (Se.3) ⊕Hash (Se.3, Se.4))
InvSbox (Se.1 ⊕MRC (Se.4) ⊕MCR (Se.4) ⊕Hash (Se.3, Se.4))
R =N?
Decrypted Image Output

Figure 14: Decryption Process

3.1.2.4 Application of the algorithm for color images

A color image is composed of three main components, i.e. R, G and B. The matrixes R, G and B of the color image are encrypted in parallel and similar to the encryption of gray level image. Decryption process of matrixes R, G and B is similar to the purposed decryption process above.

3.1.2.5 Parallel mode and its requirement for image encryption

Processing speed improvements and parallel image encryption, need elimination of the conventional CBC-like mode. However, this will cause a new problem, i.e. how to provide the diffusion requirement without such mode. Besides, parallel image encryption has some additional requirements:

(1) Computational load balancing: Slowest Processing Element (PE) determines the total time of a parallel image encryption scheme, therefore other PEs has to wait until such PE finishes its operation.

(2) Communication load balancing

(3) Critical area management: Reading and writing the same area of memory by different PEs may cause unexpected program execution. Hence it is necessary to use some parallel techniques to manage critical areas.

The proposed algorithm structure satisfies the above requirements. As it is shown in Fig. 15, the image is equally divided into 4 parts. Each of these parts is encrypted by one single PE. All of the four PEs do their encryption simultaneously. Then two distinct pairs of PEs are selected and each one uses one half of image data for encryption of the other half of the image. Generated masks in each PEs will enable our algorithm to satisfy the diffusion requirements.

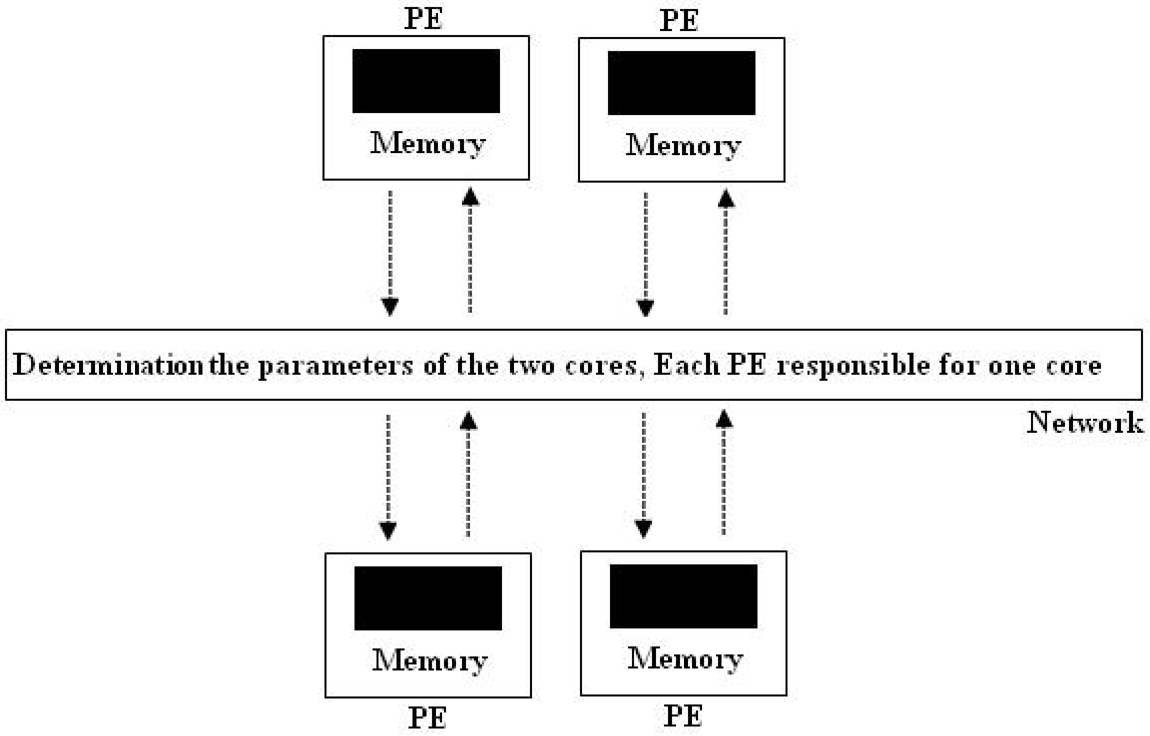

PE
PE
Memory
Memory
Determination the parameters of the two cores, Each PE responsible for one core
Network
Memory
Memory
PE
PE

Figure 15: Parallel Computation mode for image encryption

3.1.3 Performance and Security Analysis

We have done several tests to check the security of the proposed cryptosystem. Statistical tests include histogram analysis and calculation of correlation coefficients of adjacent pixels. We have done our experimental analysis for the proposed encryption scheme on the USC-SIPI image database. In this paper we have selected one gray level image ’Lena’ and one color image ’Pepper’ with 256×256 pixels as the sample plain images. Their plain, cipher and decrypted images are shown in Fig 16.

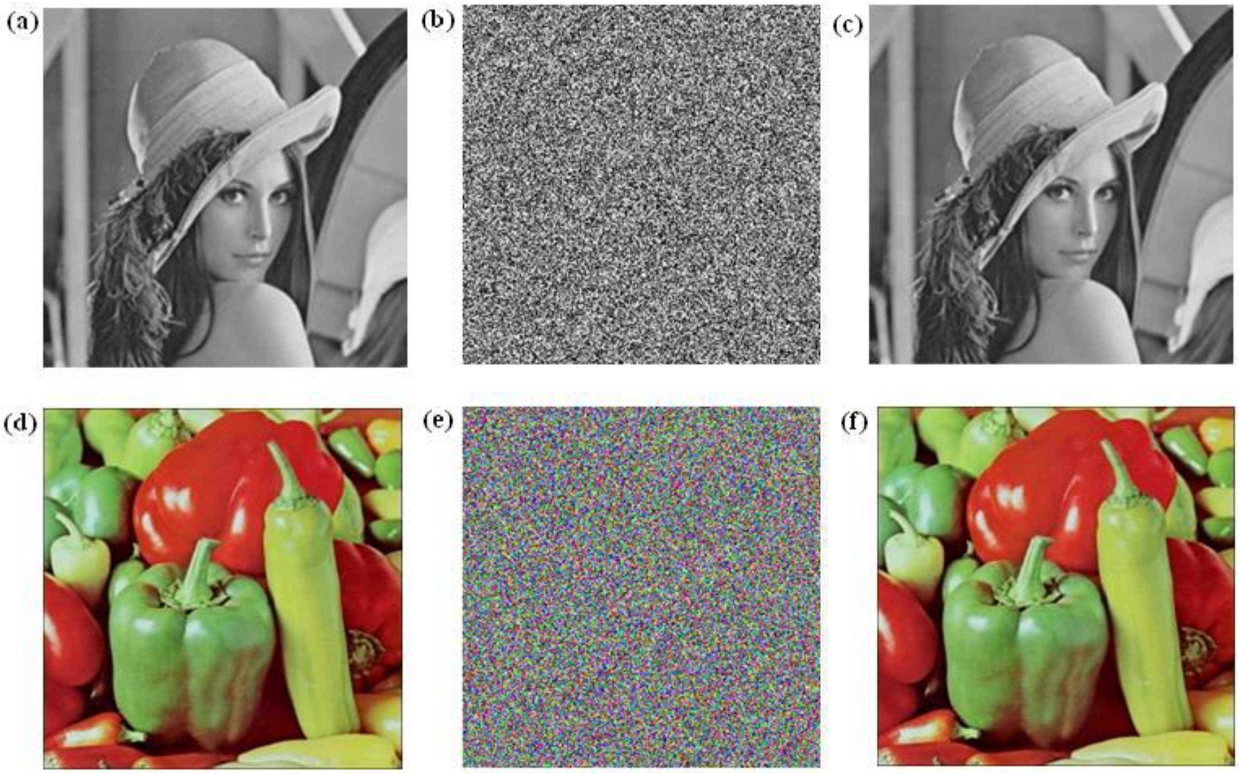


Figure 16: The plain, cipher, and decrypted images. (a) Plain-image Lena, (b)) cipher image, 2-round, (c) the decrypted image, (d) Plain-image Pepper, (e) cipher image, 2-round (f) the decrypted image

3.1.3.1 Histogram

Image histogram is a very important feature in image analysis. From Fig. 16 it is obvious that the histograms of the encrypted image are nearly uniform and significantly different from the histograms of the original image. Hence it does not provide any clue to employ any statistical analysis attack on the encryption image. Histograms of the plain and the cipher images are depicted in Fig. 17 by choosing two rounds (R=2).

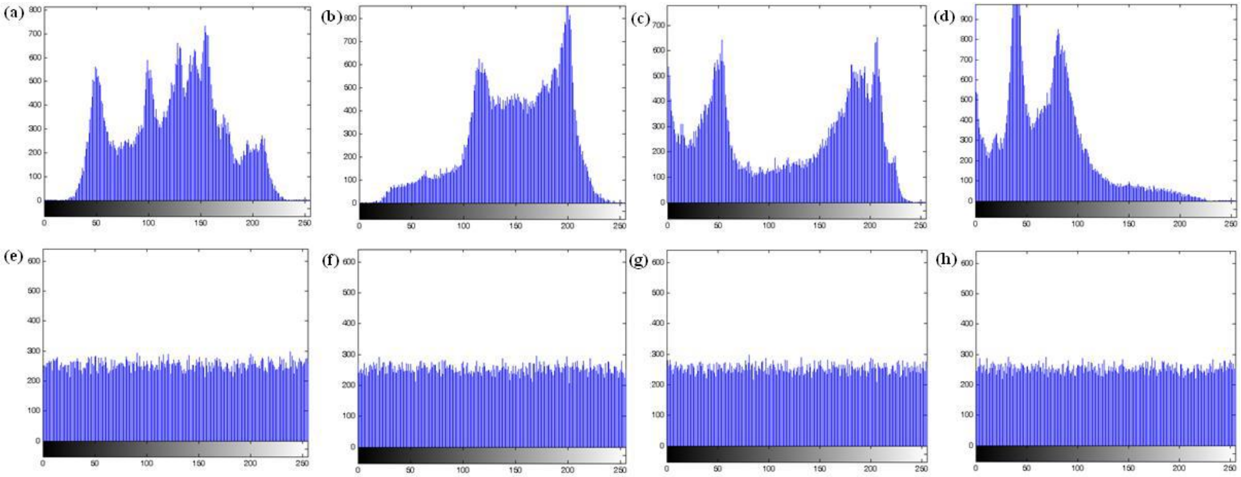


Figure 17: (a) Histogram of the plain-image Lena (b) histogram of the plain-image Pepper- R (c) histogram of the plain-image Pepper- G (d) histogram of the Pepper- B (e) histogram of the cipher image Lena (f) histogram of the cipher -image Pepper- R (g) histogram of the cipher -image Pepper- G (h) histogram of the cipher -image Pepper- B.

3.1.3.2 Correlation analysis of two adjacent pixels

We have analyzed the correlation between two vertically adjacent pixels, two horizontally adjacent pixels, and two diagonally adjacent pixels in an image. 2000 pairs of two adjacent (in vertical, horizontal, and diagonal direction) pixels from plain-image and ciphered image were randomly selected and the correlation coefficients were calculated by using the following equations:

$$r_{xy} = \frac{\|Cov(x,y)|}{\sqrt{D(x)}\sqrt{D(y)}}, \quad E = \frac{1}{N}\sum_{i=1}^{N} x_i$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))^2$$

$$Cov(x,y) = \frac{1}{N}\sum_{i=1}^{N}(x_i - E(x))(y_i - E(x))$$

x and y represent gray level values of two adjacent pixels. Table 1 is the horizontal, vertical and diagonal relevance of adjacent elements in image before and after encryption. Fig 18 shows the results of correlation analysis. Fig 18 show significant reduction in relevance of adjacent elements.

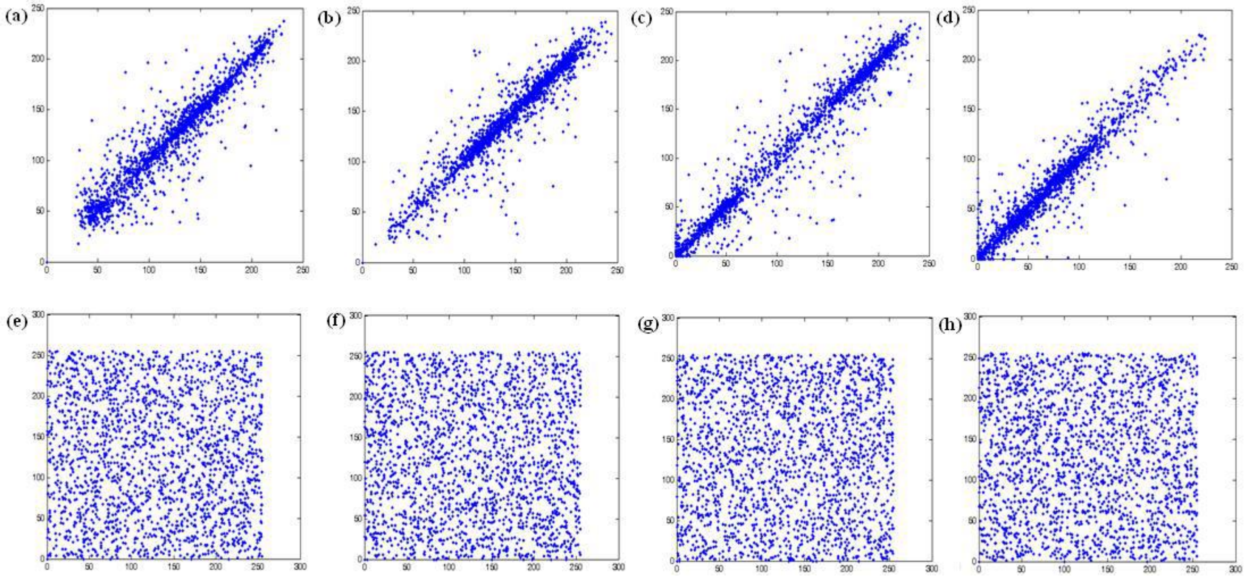


Figure 18: Correlation analysis of two horizontally adjacent pixels in (a) Plain Lena image, (b) Plain-image Pepper- R, (c) Plain-image Pepper- G, (d) Plain-image Pepper- B, (e) cipher image Lena, (f) cipher image Pepper- R, (g) cipher image Pepper- G; (h) cipher image Pepper -B ;obtained using the proposed scheme

Table 1: Correlation Coefficient of plain image and ciphered image

	Lena		Pepper					
	Plain-Image	Cipher Image	Plain-Image			Cipher Image		
Scan Direction			R	G	B	R	G	B
Horizontal	0.9491	-0.0006	0.9604	0.9733	0.9561	0.0069	-0.0019	-0.0015
Vertical	0.9768	-0.0030	0.9674	0.9796	0.9573	0.0024	0.0033	0.0096

Diagonal	0.9304	0.0061	0.9327	0.9608	0.9195	0.0001	0.0067	-0.0024

The comparison given in Table II shows that the proposed method outperforms other reported methods in the paper. The encrypted version under this scheme has the maximum performance in the horizontal and vertical directions as well as diagonal direction.

Table 2 : Performance analysis of proposed method with recent methods

Scan Direction	**Horizontal**	**Vertical**	**Diagonal**
Plain-Image Lena	0.9506031	0.9761479	0.9254270
Proposed	-0.0006	-0.0030	0.0061
Ref. [10]	-0.04005	0.08158	-0.00471
Ref. [11]	0.00534	0.00846	0.00355
Ref. [12]	0.00681	0.00782	0.00323
Ref. [23]	0.01270	-0.01900	-0.00120
Plain-Image Pepper	0.9681485	0.9616387	0.9396524
Proposed	0.00401	0.00863	0.00253
Ref. [29]	0.00070	0.00216	0.01488
Ref. [30]	0.01183	0.00016	0.01480

3.2 Conclusion

In this paper, a novel algorithm for image encryption based on SHA-512 is presented. The proposed algorithm binds together the excellent qualities of permutation and diffusion properties in four steps each round. Because of intentional removal of certain modes in this algorithm, the image encryption can now be applied in parallel. The decryption procedure is similar to that of the encryption but the direction is inverted.

REFERENCES

1. Daemen, Joan, and Vincent Rijmen. "The design of {Rijndael}:{AES}---the Advanced." Journal of Cryptology 4.1 (1991): 3-72.
2. Zeghid, Medien, et al. "A modified AES based algorithm for image encryption." International Journal of Computer Science and Engineering 1.1 (2007): 70-75.
3. Subramanyan, B., Vivek M. Chhabria, and TG Sankar Babu. "Image Encryption Based on AES Key Expansion." Emerging Applications of Information Technology (EAIT), 2011 Second International Conference on. IEEE, 2011.
4. Gong-bin, Qian, Jiang Qing-feng, and Qiu Shui-sheng. "A new image encryption scheme based on DES algorithm and Chua's circuit." Imaging Systems and Techniques, 2009. IST'09. IEEE International Workshop on. IEEE, 2009.
5. Diffie, Whitfield, and Martin E. Hellman. "New directions in cryptography." Information Theory, IEEE Transactions on 22.6 (1976).
6. Shuihua, Han, and Yang Shuangyuan. "An asymmetric image encryption based on matrix transformation." ECTI Transactions on Computer and Information Technology 1.2 (2005).
7. Ganesan, Kannabiran, Ishan Singh, and Mansi Narain. "Public key encryption of images and videos in real time using chebyshev maps." Computer Graphics, Imaging and Visualisation, 2008. CGIV'08. Fifth International Conference on. IEEE, 2008.
8. Gupta, Kamlesh, et al. "An ethical way of image encryption using ECC." Computational Intelligence, Communication Systems and Networks, 2009. CICSYN'09. First International Conference on. IEEE, 2009.

9. M. Naor and A. Shamir, Visual cryptography, Advances in Cryptology-EUROCRYPT94, Lecture Notes in Computer Science, Vol. 950, Springer- Verlag, Berlin, pp. 1-12, 1995.
10. Jaafar, Abdullah M., and Azman Samsudin. "A New Public-Key Encryption Scheme Based on Non-Expansion Visual Cryptography and Boolean Operation." IJCSI (2010).
11. Matthews, Robert. "On the derivation of a “chaotic” encryption algorithm." Cryptologia 13.1 (1989).
12. B. Mohammed, G. Mourad, Z. Nourddine, R. Fakhita and B. ElHoussine, Encryption-Compression Method of Images, Int. Journal on Computer Science and Information Systems Vol. 4, No. 1, pp. 30-41, 2009.
13. Vorwerk, Lutz, Thomas Engel, and Christoph Meinel. "Proposal for a combination of compression and encryption." Visual Communications and Image Processing 2000. International Society for Optics and Photonics, 2000.
14. Masanori, Ito, et al. "New Image Encryption and Compression Method Based on Independent Component Analysis." Information and Communication Technologies: From Theory to Applications, 2008. ICTTA 2008.
15. Wu, Chung-Ping, and C-CJ Kuo. "Design of integrated multimedia compression and encryption systems." Multimedia, IEEE Transactions on 7.5 (2005): 828-839.
16. Effelsberg, Wofgang, and Ralf Steinmetz. Video Compression Techniques: From JPEG to Wavelets. Vol. 1. Morgan Kaufmann Pub, 1998.
17. Zeng, Wenjun, and Shawmin Lei. "Efficient frequency domain selective scrambling of digital video." Multimedia, IEEE Transactions on 5.1 (2003): 118-129.

18. Spanos, George A., and Tracy Bradley Maples. "Performance study of a selective encryption scheme for the security of networked, real-time video." icccn. Published by the IEEE Computer Society, 1995.

19. Agi, Iskender, and Li Gong. "An empirical study of secure MPEG video transmissions." Network and Distributed System Security, 1996., Proceedings of the Symposium on. IEEE, 1996.

20. Van Droogenbroeck, Marc, and Raphaël Benedett. "Techniques for a selective encryption of uncompressed and compressed images." ACIVS Advanced Concepts for Intelligent Vision Systems, Proceedings (2002).

21. Van Droogenbroeck, Marc. "Partial encryption of images for real-time applications." Fourth IEEE Benelux Signal Processing, Hilvarenbeek, The Netherlands (2004): 11-15.

22. Odibat, Omar M., Moussa H. Abdallah, and Moh'D. Belal R. Al-Zoubi. "New Techniques in the Implementation of the Partial Image Encryption." 2006.

23. Hong, Kwangjin, and Keechul Jung. "Partial encryption of digital contents using face detection algorithm." PRICAI 2006: Trends in Artificial Intelligence. Springer Berlin Heidelberg, 2006. 632-640.

24. Rodrigues, J. M., et al. "Face protection by fast selective encryption in a video." (2006): 420-425.

25. Lian, Shiguo, et al. "A selective image encryption scheme based on JPEG2000 codec." Advances in Multimedia Information Processing-PCM 2004. Springer Berlin Heidelberg, 2005. 65-72.

26. Taneja, Nidhi, Balasubramanian Raman, and Indra Gupta. "Selective image encryption in fractional wavelet domain." AEU-International Journal of Electronics and Communications 65.4 (2011): 338-344.
27. Lorenz, Edward N. The essence of chaos. University of Washington Press, 1995.
28. Kellert, Stephen H. In the wake of chaos: Unpredictable order in dynamical systems. University of Chicago press, 1994.
29. Kocarev, Ljupco. "Chaos-based cryptography: a brief overview." Circuits and Systems Magazine, IEEE 1.3 (2001): 6-21.
30. Rivest, Ronald L., Adi Shamir, and Leonard Adleman. "A method for obtaining digital signatures and public-key cryptosystems." Communications of the ACM 26.1 (1983).
31. Lin, Ching-Yung, and Shih-Fu Chang. "Generating robust digital signature for image/video authentication." Multimedia and Security Workshop at ACM Multimedia. Vol. 98. 1998.
32. Chen, Tao, Jingchun Wang, and Yonglei Zhou. "Combined digital signature and digital watermark scheme for image authentication." Info-tech and Info-net, 2001. Proceedings. ICII 2001-Beijing. 2001 International Conferences on. Vol. 5. IEEE, 2001.
33. Lu, Chun-Shien, and H-YM Liao. "Structural digital signature for image authentication: an incidental distortion resistant scheme." Multimedia, IEEE Transactions on 5.2 (2003): 161-173.
34. Zang, Hongyan, Lequan Min, and Li Cao. "An Image Encryption and Digital Signature Scheme Based on Generalized Synchronization Theorem." Computational Intelligence and Security, 2009. CIS'09. International Conference on. Vol. 1. IEEE, 2009.

35. Zhao, Gaochang, et al. "RSA-based digital image encryption algorithm in wireless sensor networks." Signal Processing Systems (ICSPS), 2010 2nd International Conference on. Vol. 2. IEEE, 2010.
36. K. Gupta, S. Silakari, Performance Analysis for Image Encryption Using ECC, Int. Conference on Computational Intelligence and Communication Networks, 2010.
37. Yahya, Abdelfatah A., and Ayman M. Abdalla. "A shuffle image-encryption algorithm." Journal of Computer Science 4.12 (2008): 999.
38. Rodrigues, José M., William Puech, and Adrian G. Bors. "Selective encryption of human skin in JPEG images." Image Processing, 2006 IEEE International Conference on. IEEE, 2006.
39. http://en.wikipedia.org/wiki/Hash_function
40. Sobti, Rajeev, and G. Geetha. "Cryptographic Hash Functions: A Review." IJCSI International Journal of Computer Science Issues 9.2 (2012): 461-479.
41. AlAhmad, Mohammad A., and Imad Fakhri Alshaikhli. "Broad View of Cryptographic Hash Functions." (2013).
42. Alkandari, Abdulaziz Ali, Imad Fakhri Al-Shaikhli, and Mohammad A. Alahmad. "Cryptographic Hash Function: A High Level View." Informatics and Creative Multimedia (ICICM), 2013 International Conference on. IEEE, 2013.
43. Al-Kuwari, Saif, James H. Davenport, and Russell J. Bradford. "Cryptographic hash functions: recent design trends and security notions." (2010): 133-150.
44. S. Lucks, “Design Principled for Iterated Hash Functions”, in IACR Cryptology ePrint Archive, 2004, pp. 253.

45. M. Bellare, and T. Kohno, "Hash Function Balance and Its Impact on Birthday Attacks", in EUROCRYPT, 2004, pp.401-418.

46. Seyedzade, S.M.; Mirzakuchaki, S.; Atani, R.E., "A novel image encryption algorithm based on hash function," Machine Vision and Image Processing (MVIP), 2010 6th Iranian , vol., no., pp.1,6, 27-28 Oct. 2010

47. Gauravaram, Praveen, William Millan, and Lauren May. "CRUSH: A New Cryptographic Hash Function using Iterated Halving Technique." Cryptographic Algorithms and their Uses. 2004.

48. R. Glabb, L. Imbert and G. Jullien, Multi-mode operator for SHA-2 hash functions, Journal of Systems Architecture , vol. 53, no. 2-3, pp. 127-138, 2007.

49. Bagheri, Nasour, Majid Naderi, and Babak Sadeghiyan. "Cryptanalysis of CRUSH hash structure." IACR Cryptology ePrint Archive 2008 (2008)

50. Bagheri, N.; Henricksen, M.; Knudsen, L.R.; Naderi, M.; Sadeghyian, B., "Cryptanalysis of an iterated halving-based hash function: CRUSH," Information Security, IET , vol.3, no.4, pp.129,138, Dec. 2009

51. Chia-Yu Lu; You-Wei Lin; Shang-Ming Jen; Jar-Ferr Yang, "Cryptanalysis on PHOTON hash function using cube attack," Information Security and Intelligence Control (ISIC), 2012 International Conference on , vol., no., pp.278,281, 14-16 Aug. 2012

52. Norouzi, Benyamin, et al. "A novel image encryption based on hash function with only two-round diffusion process." Multimedia Systems 20.1 (2014): 45-64.

53. Sklavos, N., "Multi-module Hashing System for SHA-3 & FPGA Integration," Field Programmable Logic and Applications (FPL), 2011 International Conference on , vol., no., pp.162,166, 5-7 Sept. 2011