ROBUST ENCRYPTION SCHEMES
FOR 3D CONTENT PROTECTION

ALIREZA JOLFAEI
BSc.Eng., MSc.Eng.

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY
GRIFFITH SCIENCES, GRIFFITH UNIVERSITY
QUEENSLAND, AUSTRALIA

Submitted in fulfilment
of the requirements of the degree of
DOCTOR OF PHILOSOPHY

November 2015
Abstract

Since the 1970s, a large number of encryption schemes have been proposed, among which some have been standardised and widely adopted all over the world, such as data encryption standard (DES) and advanced encryption standard (AES). However, due to the special features of three-dimensional (3D) content, these encryption standards are not a suitable solution for 3D applications. The problem of 3D content encryption is beyond the application of established and well-known encryption algorithms. This is primarily due to the structure of 3D content and the way it is used commercially. Unlike data encryption, where a complete bitstream is encrypted, 3D content encryption introduces several challenges. One of the greatest challenges of 3D content encryption is that, in comparison with traditional data and 2D images, 3D content implies a higher level representation or semantics, and in many 3D applications, it is necessary to maintain 3D semantics, such as the spatial and dimensional stability. The major aim of this thesis is to investigate innovative solutions for encrypting 3D content which ensures the usability of encrypted content through maintaining the spatial and dimensional semantics. To this end, we overviewed the relevant background of 3D content and data encryption. We also investigated the limitations of the current techniques in addressing the challenges of 3D content encryption. The literature review delineated the scope of the research and identified the existing problems and limitations. The important research problems in the field of 3D content encryption are summarised as follows: i) the current encryption technology is not able to preserve the dimensional and spatial stability of 3D content; ii) the correlation between the texture image information and 3D surface geometry discloses the hidden geometry of protected 3D objects; iii) direct application of encryption algorithms to 3D content requires high computational power and introduces delay in real-time communication; and iv) there are no design principles for the construction of secure cryptographic primitives for 3D content, that is, those which are resistant to well-known attacks such as surface reconstruction attacks. To solve these research problems, we investigated several existing 2D and 3D encryption schemes in which we found several security flaws. These cryptanalyses helped us in designing resistant ciphers against well-known attacks. In addition, as efficiency is an important concern in 3D applications, we investigated efficient and fast primitives for 3D content encryption. We established a new mathematical notion which ensures the usability of encrypted 3D content in 3D applications by maintaining the dimensional and spatial stability of encrypted 3D content. We proposed the first solution, which is both theoretically and practically important, for the design and evaluation of an appropriate 3D content encryption scheme. Theoretical and experimental analyses demonstrated the effectiveness of our design principles. The design approach is convincingly supported by a number of new cryptanalytic results. Furthermore, the results of our rigorous cryptanalysis can be used in the future design and analysis of complex encryption schemes for multimedia applications.
This work has not previously been submitted for a degree or diploma in any university. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is made in the thesis itself.

(Signed)  ———————

Alireza Jolfaei

November 2015
This thesis is dedicated to my parents.
For their endless love, support and encouragement.
Acknowledgements

When I first started my PhD, I did not know where it would take me or even how much I would achieve. I have been incredibly lucky to have worked on some interesting problems which have given birth to significant results. This has all been possible due to the excellent supervision of Dr Xin-Wen Wu and Associate Professor Vallipuram Muthukkumarasamy. Their advice, constructive criticism and the odd amusing analogy have helped me, not only to develop as a researcher but have taught me a great deal about writing and presenting my work. What I have learnt will be invaluable in my future career.

I am grateful to Professor Xun Yi, Professor Yi Mu, Professor Willy Susilo, Professor Josef Pieprzyk and the members of the Network Security Research Group at Griffith University for their insightful and invaluable comments which have helped me significantly improve my work.

I am also grateful to Griffith University for the research scholarship and financial support during my PhD program which has kept me well fed and watered throughout the course of my PhD.

During my time at Griffith University I have been lucky to meet and become friends with a vast array of people. There have been many great times shared in the office, on the athletic court or at the coffee shop. Thanks to everyone who has made my time at Griffith University so enjoyable.

Finally, I would like to thank my parents for always being able to put a smile on my face and for their unwavering support and guidance.
List of Publications

Parts of this thesis have already been published in a number of international refereed journals and conference proceedings. The published papers are cited appropriately throughout this thesis and are listed below:

   ▶ The work in this paper is presented in Chapter 6.

   ▶ The work in this paper is presented in Chapter 4. This work has been recognised as the current leading research by the IEEE Australian Council, and this paper has received the prestigious IEEE Australian Council Runner-Up Prize for the best postgraduate student paper in 2015.

   ▶ The work in this paper is presented in Chapter 2.

   ▶ The work in this paper is presented in Chapter 6.

   ▶ The work in this paper is presented in Chapter 5.

   ▶ The work in this paper is presented in Chapter 4. I received the competitive IEEE signal processing society conference travel grant as well as the Griffith graduate research school
(GGRS) and international experience incentive scheme (IEIS) conference travel grants to attend this conference.


◻ The work in this paper is presented in Chapter 4.

Table of Contents

Abstract ii

Declaration v

Acknowledgements viii

List of Publications ix

Table of Contents xi

List of Figures xv

List of Tables xvii

Abbreviations xix

Notations xxiii

1 Introduction 1

1.1 Motivation 1

1.2 Privacy-Sensitive Applications 3

1.3 Research Contribution 5

1.4 Organisation of the Thesis 6

2 Background and State of the Art in 3D Content Encryption Schemes 9

2.1 Introduction 9

2.2 3D Content 10

2.2.1 Image 10

2.2.1.1 Texture Map 11

2.2.2 3D Models 11

2.2.2.1 Creation of 3D Models 12

2.2.3 Compression 13

2.2.4 3D Content Formats 14

2.2.5 Virtual Worlds 16

2.2.6 Rendering Pipeline 17

2.2.7 Virtual Camera 17

2.3 Data Security 18

2.3.1 3D Content Security 20

2.3.2 Cryptology 20

2.3.2.1 Cryptography 21

2.3.2.2 Cryptanalysis 24
## List of Figures

2.1 Different representations of *Victoria21*: (a) a cloud of points, (b) a triangular mesh, (c) a set of parametric surfaces, (d) a set of voxels. .......................... 12

2.2 Sources of 3D objects. .................................................. 13

2.3 Graphics pipeline. ....................................................... 17

2.4 Camera view plane. ...................................................... 18

2.5 Maslow’s hierarchy of needs ......................................... 19

2.6 Data confidentiality big picture .................................... 21

2.7 General setting of private communication ....................... 21

2.8 Stages taken during naïve and selective encryption approaches. .......................... 37

2.9 Communication model of 3D content. ............................. 47

3.1 Block diagram of the research process. .......................... 54

3.2 Diagrammatic outline of the methodology. ....................... 58

4.1 Structure of the encryption algorithm. .......................... 69

4.2 Maximal space that a random rotation can disperse a point. .......................... 76

4.3 Boundary of the rotated point cloud. ............................. 77

4.4 Similarity analysis: (a) the plain point cloud, (b) the cipher point cloud, (c) the pairwise Euclidean distance between the corresponding points of the plain point cloud and cipher point cloud, (d) the Hausdorff distance between the plain point cloud and cipher point cloud, (e) the heat map of the distance matrix. .......................... 82

4.5 Plaintext sensitivity test result: (a) original plain point cloud, (b) encrypted point cloud from the original plaintext, (c) slightly changed plain point cloud by displacing 1 point by a distance 0.1% of radius $r$ of the bounding sphere, and (d) encrypted point cloud from the slightly changed plaintext. .......................... 84

4.6 Pairwise Euclidean distance between the corresponding points of (a) plain point clouds and (b) encrypted point clouds. .......................... 85

4.7 Log-log plot of the Hausdorff distance against the ratio of the spatial displacement. .......................... 85

4.8 Key sensitivity test result: (a) plain point cloud *Centaur5* \[1\], (b) encrypted point clouds using the original and slightly modified secret keys, and (c) Euclidean distance between points of two encrypted point clouds. .......................... 86

4.9 Sampling distribution of $Z$-values. .............................. 90

4.10 Surface reconstruction using the normal analysis, (a) plain point cloud, (b) reconstructed surface from the cipher point cloud, and (c) refined reconstructed surface using the normal analysis. .......................... 92

4.11 The encryption time of various sized point clouds. .......................... 94

5.1 (a) A zigzag path to scramble bits of a bit-plane image, (b) Permutation result of the bit-plane image for mod ($s$, 12) = 7. .......................... 99
5.2 Encryption results of a sample texture image: (a) original image, (b) encrypted image using full AES, (c) encrypted image using selective AES, (d) encrypted image using Salsa Dance.

5.3 Correlation analysis and distribution of two adjacent pixels in the plain-image and cipher-image.

5.4 Results of edge detection by median and Canny filtering for the (a) encrypted image using full AES, (b) encrypted image using selective AES, and (c) encrypted image using Salsa Dance.

6.1 Construction procedure of the chosen plain-image/cipher-image pairs for $M = N = L = 2$.

6.2 Test images used in the experiments.

6.3 Corresponding cipher-images of the six test images.

6.4 Required pairs of chosen input/output images (a) with size $256 \times 256$ for finding the permutation matrix of size $256 \times 256$, and (b) with size $512 \times 512$ for finding the permutation matrix of size $512 \times 512$.

6.5 Decrypted images of the (a) cipher-image #1 and (b) cipher-image #4.

6.6 Corresponding encrypted bit-plane images of the six test bit-plane images.

6.7 Required pairs of chosen input/output bit-plane images with size $256 \times 2048$ for finding the permutation matrix of size $256 \times 2048$.

6.8 Decrypted images of the (a) cipher-image #2 and (b) cipher-image #5.

6.9 Percentage of correctly recovered elements with respect to the number of chosen plain-images.

6.10 One-round combination of a permutation primitive with a substitution primitive.

6.11 Pareek et al.'s encryption scheme.

A.1 A number of the used texture images.

A.2 A number of the used 3D objects.
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Comparison of the multimedia-scene description standards</td>
<td>15</td>
</tr>
<tr>
<td>2.2</td>
<td>Summary of the recent work in image encryption</td>
<td>38</td>
</tr>
<tr>
<td>2.3</td>
<td>Comparison among 3D content encryption methods</td>
<td>48</td>
</tr>
<tr>
<td>4.1</td>
<td>Comparison of key space</td>
<td>78</td>
</tr>
<tr>
<td>5.1</td>
<td>Comparison of the relative CPU time</td>
<td>101</td>
</tr>
<tr>
<td>5.2</td>
<td>Correlation coefficients of two adjacent pixels in plain-image and cipher-image</td>
<td>105</td>
</tr>
<tr>
<td>5.3</td>
<td>2D correlation coefficients between the RGB colour layers of the cipher-images</td>
<td>106</td>
</tr>
<tr>
<td>5.4</td>
<td>Comparison of the PSNR values</td>
<td>106</td>
</tr>
<tr>
<td>6.1</td>
<td>Performance test</td>
<td>126</td>
</tr>
<tr>
<td>6.2</td>
<td>The first 16 elements of the retrieved plain-image $P$</td>
<td>130</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>2D</td>
<td>Two-Dimensional</td>
<td></td>
</tr>
<tr>
<td>3D</td>
<td>Three-Dimensional</td>
<td></td>
</tr>
<tr>
<td>AC</td>
<td>Alternating Current</td>
<td></td>
</tr>
<tr>
<td>ACM</td>
<td>Arnold Cat Map</td>
<td></td>
</tr>
<tr>
<td>ADSL</td>
<td>Asymmetric Digital Subscriber Line</td>
<td></td>
</tr>
<tr>
<td>AES</td>
<td>Advanced Encryption Standard</td>
<td></td>
</tr>
<tr>
<td>ANSI</td>
<td>American National Standards Institute</td>
<td></td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
<td></td>
</tr>
<tr>
<td>ASIO</td>
<td>Australian Security Intelligence Organisation</td>
<td></td>
</tr>
<tr>
<td>BPP</td>
<td>Bits Per Pixel</td>
<td></td>
</tr>
<tr>
<td>CAD</td>
<td>Computer-Aided Design</td>
<td></td>
</tr>
<tr>
<td>CAGR</td>
<td>Compound Annual Growth Rate</td>
<td></td>
</tr>
<tr>
<td>CBC</td>
<td>Cipher Block Chaining</td>
<td></td>
</tr>
<tr>
<td>CKBA</td>
<td>Chaotic Key-Based image encryption Algorithm</td>
<td></td>
</tr>
<tr>
<td>CLT</td>
<td>Central Limit Theorem</td>
<td></td>
</tr>
<tr>
<td>CML</td>
<td>Coupled Map Lattice</td>
<td></td>
</tr>
<tr>
<td>COLLADA</td>
<td>COLLABorative Design Activity</td>
<td></td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
<td></td>
</tr>
<tr>
<td>CryptAPI</td>
<td>Cryptographic Application Programming Interface</td>
<td></td>
</tr>
<tr>
<td>CT</td>
<td>Computed Tomography</td>
<td></td>
</tr>
<tr>
<td>DC</td>
<td>Direct Current</td>
<td></td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete Cosine Transform</td>
<td></td>
</tr>
<tr>
<td>DES</td>
<td>Data Encryption Standard</td>
<td></td>
</tr>
<tr>
<td>DRM</td>
<td>Digital Rights Management</td>
<td></td>
</tr>
<tr>
<td>DWT</td>
<td>Discrete Wavelet Transform</td>
<td></td>
</tr>
</tbody>
</table>
ECMA European Computer Manufacturers Association
FIPS Federal Information Processing Standard
GB Giga Byte
GPU Graphics Processing Unit
IBM International Business Machines
IEC International Electrotechnical Commission
IID Independent and Identically Distributed
IP Intellectual Property
ISO International Standardisation Organisation
IV Initialisation Vector
JPEG Joint Photographic Experts Group
JPSEC JPEG 2000 Secured
LSB Least Significant Bit
MMSP MultiMedia Signal Processing
MPEG Moving Picture Experts Group
MRI Magnetic Resonance Imaging
MSB Most Significant Bit
MSE Mean Squared Error
NBS National Bureau of Standards
NIST National Institute of Standards and Technology
NSW New South Wales
OBJ Wavefront OBJect file format
P2P Peer-To-Peer
PDF Portable Document Format
PLY PoLYgon file format
PRNG Pseudo-Random Number Generator
PSNR Peak Signal-to-Noise Ratio
ROI Region Of Interest
RSA Rivest, Shamir, Adleman
SIPI Signal and Image Processing Institute
SPIHT Set Partitioning In Hierarchical Trees
SPN Substitution-Permutation Network
SRTP Secure Real-time Transport Protocol
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STL</td>
<td>STereoLithography</td>
</tr>
<tr>
<td>TDCEA</td>
<td>Two-Dimensional Circulation Encryption Algorithm</td>
</tr>
<tr>
<td>U3D</td>
<td>Universal 3D</td>
</tr>
<tr>
<td>USC</td>
<td>University of Southern California</td>
</tr>
<tr>
<td>VLSI</td>
<td>Very Large Scale Integration</td>
</tr>
<tr>
<td>VRML</td>
<td>Virtual Reality Modelling Language</td>
</tr>
<tr>
<td>WG</td>
<td>Working Group</td>
</tr>
<tr>
<td>X3D</td>
<td>eXtensible 3D</td>
</tr>
<tr>
<td>XML</td>
<td>eXtensible Markup Language</td>
</tr>
</tbody>
</table>
The main notations used throughout this thesis are listed below. Vectors are denoted in bold.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x \oplus y$</td>
<td>Addition of $x$ and $y$ modulo 2 (XOR)</td>
</tr>
<tr>
<td>$\lfloor x \rfloor$</td>
<td>The largest integer not greater than $x$</td>
</tr>
<tr>
<td>$\lceil x \rceil$</td>
<td>The smallest integer not less than $x$</td>
</tr>
<tr>
<td>$x \mid y$</td>
<td>$y$ is divisible by $x$</td>
</tr>
<tr>
<td>$A \cup B$</td>
<td>The union of two sets $A$ and $B$</td>
</tr>
<tr>
<td>$A \cap B$</td>
<td>The intersection of two sets $A$ and $B$</td>
</tr>
<tr>
<td>$a \mod n$</td>
<td>The remainder of the Euclidean division of $a$ by $n$</td>
</tr>
<tr>
<td>$n!$</td>
<td>The factorial of a non-negative integer $n$, that is, $\prod_{k=1}^{n} k$. By convention $0! = 1$.</td>
</tr>
<tr>
<td>$u \ll l$</td>
<td>The $l$-bit left rotation of a word $u$.</td>
</tr>
<tr>
<td>$0b$</td>
<td>The value which follows is in binary notation.</td>
</tr>
<tr>
<td>$0x$</td>
<td>The value which follows is in hexadecimal notation.</td>
</tr>
<tr>
<td>$\inf S$</td>
<td>The infimum of a subset $S$ of a partially ordered set $T$ is the greatest element in $T$ that is less than or equal to all elements of $S$, if such an element exists. The infimum is also referred to as the greatest lower bound.</td>
</tr>
<tr>
<td>$\sup S$</td>
<td>The supremum of a subset $S$ of a totally or partially ordered set $T$ is the least element in $T$ that is greater than or equal to all elements of $S$, if such an element exists. The supremum is also referred to as the least upper bound.</td>
</tr>
<tr>
<td>$\gamma(t)$</td>
<td>The gamma function is defined for all complex numbers except the non-positive integers. For complex numbers with a positive real part, it is defined via a convergent improper integral, that is, $\int_{0}^{\infty} x^{t-1} e^{-x} dx$.</td>
</tr>
<tr>
<td>$N(0, 1)$</td>
<td>Standard normal distribution with mean $\mu = 0$ and variance $\sigma = 1$</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Mean of Poisson process</td>
</tr>
<tr>
<td>$\mathbb{R}$</td>
<td>The set of real numbers</td>
</tr>
<tr>
<td>$P$-value</td>
<td>The probability of obtaining a test statistic at least as extreme as the one that was actually observed, assuming that the null hypothesis is true</td>
</tr>
<tr>
<td>$Pr(x)$</td>
<td>Probability of observing event $x$</td>
</tr>
<tr>
<td>$\Phi(x)$</td>
<td>Cumulative distribution function of a random variable $x$</td>
</tr>
</tbody>
</table>
"The more you know, the more you realise you know nothing.”
– Socrates, 470 BC – 399 BC.

1

Introduction

DATA protection, which has for a long time fascinated many people, continues to be a fundamental problem in cryptography. This problem has become even more important with recent advances in 3D computing and 3D modelling, which have unlocked the path for the application of 3D content in a variety of domains, including virtual reality and augmented reality. The problem of 3D content protection is beyond the application of established and well-known protection methods. This is primarily due to the structure of 3D content and the way it is used commercially. 3D content protection involves various aspects, including authentication, confidentiality, access control, and copyright protection. In this thesis, our aim is to investigate and propose technical solutions for preserving confidentiality of 3D content using encryption methods.

1.1 Motivation

Over the past decade, there has been a huge rise in usage and distribution of 3D content in various industries. It is anticipated that the market for 3D content will reach $4.4 billion by 2020 at a compound annual growth rate (CAGR) of 6.4% from 2014 to 2020 [2]. The growing applicability and revenue of 3D content suggest the necessity for protecting such assets. However, 3D assets are often not well protected because current protection methods impose too many computation overheads for data storage and transmission which delay smooth real-time rendering. Moreover, users need to do some pre-processing to gain access to protected content. In addition
to the computation overheads and accessibility problems, there is also another problem, which
the literature review in Chapter 2 reveals, that is, the 3D content semantic problem: maintaining
the dimensional and spatial stability. For most applications, maintaining the dimensional and
spatial requirements is a prerequisite for 3D content usability.

Unprotected rich and complex 3D content can be constructive, particularly in research purposes,
but it may lead to the following serious security issues:

- As there is no cost for accessing 3D content, it is easy prey for hackers. According to
  the confidential section of a Pentagon report on cyber security [3], 3D designs of more
  than two dozen of the U.S.’s most sensitive advanced weapons systems including drones,
  armour and missiles, which were the result of decades in development, have been com-
 promised by hackers. Another recent example of 3D designs being security compromised
  was the hacking of 3D design models of the A$630 million control centres of the Aus-
  tralian security intelligence organisation (ASIO) by malicious attacks on the computers of
  a construction contractor, which uncovered not only building layouts, but also the location
  of communication and computer networks [4]. These cases indicate that even top security
  agencies are still not well prepared for protecting their 3D assets.

- Unprotected content can easily be ripped and edited by unauthorised users and repro-
 cessed in other uncontrolled virtual contexts. Losing control over 3D assets annihilates
  all content production efforts and offers a free advantage to rival companies. Moreover,
  the leaking of incomplete 3D models onto the Internet can generate bad press for the stu-
  dio that created those models. These outcomes may harm the revenue of the majority of
  3D content developers, such as filming and gaming studios. A recent example is the theft
  of unfinished 3D designs of a skyscraper in Beijing [5]. The compromised 3D models
  were used to commence construction on a replica skyscraper in southwest Chongqing,
  and to add insult to injury, the unauthorised replica building was completed first.

- Unprotected content can easily be reused as a prototype to build many derivative products
  outside the provisions involving authorised creation of the 3D content. Such derivative
  products can be physical objects or simulated replicas. Physical objects are manufactured
  by transforming virtual designs into solid objects using a 3D printer, and simulated repli-
  cas are manufactured by 3D computer graphics software. A recent example is the New
  South Wales (NSW) police warning about the use of publicly available and unprotected
  3D firearm models on the Internet [6]. The NSW police demonstrated that such models
  can easily be manufactured by 3D printers and hence they can be used in serious crimes.

With regard to these security issues, it is reasonable to envisage a protection framework for
3D content and thus to review whether existing technologies are applicable in a direct manner.
3D content protection involves various aspects, including authentication, confidentiality, access
control, and copyright protection. Generally, content confidentiality, authentication and access control are addressed by encryption, through which only authorised parties holding decryption keys can access content in clear text. On the other hand, copyright protection is normally addressed by digital watermarking which embeds the owner’s private information, in what is known as a watermark, into the original content and extracts it from questionable content when the ownership needs to be resolved [7]. Watermarking and encryption are two differing techniques used for differing purposes. Encryption provides a means for secure delivery of content to the authorised party. However, encryption provides no protection once the content is decrypted. Decrypted content can easily be altered and copied by an untrustworthy user without the permission of the content owner. Therefore, to complement encryption and detect content piracy, watermarking is applied by embedding a message within the content. However, watermarking is not the focus of this research. In this thesis, our aim is to investigate 3D content encryption methods.

In addition to encryption, reliability of transmission of encrypted content over noisy communication channels, such as those found in wireless networks and the Internet with congested switches, is an important concern. If an error occurs in the encrypted data, the decryption process may not be able to reconstruct the original content, and might even propagate the error within the data. This necessitates the development of algorithms that combat bit errors or packet losses. However, maintaining the reliability (bit error correction) of 3D content transmission over noisy communication channels is not in the focus of this research. This thesis is about designing a secure and efficient crypto primitive for 3D applications, in which the rendering and displaying of 3D content is essential.

1.2 Privacy-Sensitive Applications

Contemporary 3D technologies have simplified the efficient creation of accurate 3D models of many physical objects. This has led to widespread applications of 3D content, including socialising metaverses, games, simulation tools, military affairs, medicine, chemistry, e-commerce, e-learning, e-tourism, archaeology, and cultural heritage. Although the above mentioned applications include different processes, they all have common objectives, namely, visualisation, simulation or numerical calculation. Many of the application areas require protection for 3D content. Some privacy-sensitive applications are described below.

- **Social Networks**: Human beings are social creatures and are inclined to form interest groups and exchange ideas. 3D social networks, such as cafe4tune [8] and Second Life [9], are computer-generated virtual worlds which enable individuals and organisations worldwide to interact in virtual, 2D and 3D environments. In 3D social networks, users can generate and embed their own 3D content and can interact with other contents in real time. The 3D content within such worlds is shared among the users. Such content
is vulnerable to security threats. It is therefore important to protect users’ intellectual properties (IPs).

- **Business and Commerce**: With the development of 3D technologies and the increasing demand for 3D products, many marketing firms and large companies have a considerable interest in transplanting existing real world business models into virtual worlds, and exploiting their capability in supporting 3D simulation of environments and data visualisation. For instance, IBM and Cisco recently focused on using virtual worlds to host business meetings and conferences with customers and trading partners [10]. Employees, suppliers and customers interact closely in virtual worlds, collaborating on projects by using the 3D graphical platform to visualise data and design products. The in-world companies generate revenues from the development and sale of virtual accessories, such as virtual furniture and appliances, scripted virtual tools, and avatar clothing. 3D visualisation helps customers enhance their virtual world experience. For instance, the clothing can be tried out on customised avatars. Once the avatar has the same proportions as the user and the user’s home has been modelled, the viewer can try out new furniture and see how it feels. In virtual worlds, service providers may place valuable objects to trade with end-users in specialised stores in a virtual environment. Hence, they need to protect such objects. A number of 3D content developers have reported that their customers are reluctant to buy online content due to the lack of protection [11].

- **Education and Training**: The synthetic 3D graphical environments provided by virtual worlds are particularly useful for hosting distance education, simulation of learning environments, visualisation of models, as well as classroom activities that require close interaction among students. An example is the case of virtual 3D simulators where users can practice many real world experiments by using virtual objects. Users can change their viewpoint or rotate and move objects for better viewing. In such applications, 3D simulation companies may provide access to the full scene only to paid customers. Despite such marketing strategies, unpaid customers may still need to interact and experiment with unprotected 3D objects and be able to modify the geometries. Such modifications should not affect the existing graphical elements that would be indiscernible due to the protection framework.

- **Animation and Gaming**: In animation and game production workflow, there are a number of level designers, artists and programmers who cooperate under the supervision of a producer. The output of the production workflow is an animation or a game program and a series of assets to be consumed by the animation or the game program. These assets include 3D models and images data. The 3D designs are normally updated during the production workflow. For instance, some parts or textures can be added to or subtracted from a particular object. In the production workflow, the design studio may not be willing
to give access to the full scene to a particular crew. However, this crew may need to have access to visual cues in order to insert graphical elements without striking existing ones that would be invisible because of the protection framework.

1.3 Research Contribution

The contribution of this thesis is to propose technical solutions, that is, to design and evaluate encryption schemes for preserving the confidentiality of 3D content. In this thesis, we assumed that 3D content is a combination of 3D objects and texture images. Therefore, we have only focused on addressing the security requirements of 3D objects and texture images, rather than other formats of multimedia data, such as videos, which have different security requirements. 3D content encryption is an emerging area which principally intends to preserve the functionality of 3D content. This field of security has a challenging interdisciplinary nature and faces many issues including the requirements of multimedia communication, multimedia retrieval, multimedia compression and hardware resource usage. Traditional forms of data security cannot sufficiently address the needs of 3D content security, because they mainly process information at the bit-level, which is oblivious to the semantics of the information. To overcome this difficulty, several researchers developed special encryption schemes for 3D content. However, as the literature review in Chapter 2 reveals, these encryption schemes cannot sufficiently address the technical requirements of 3D content and are insecure from the cryptographic point of view. Therefore, to overcome limitations of current techniques in addressing the security challenges of 3D content, we aim to find a solution by addressing the following main research problems.

The main problem in the field of 3D content encryption is the usability of encrypted content in 3D applications. In many 3D applications, it is necessary to maintain the location and size of the encrypted content within a defined bounding volume. If this consideration is not taken into account, the rendering result of the encrypted 3D content would, in general, overlap with other public 3D content, and this would therefore corrupt the whole virtual scene. To address this problem, we defined a mathematical notion for the dimensional and spatial stability of 3D content, and established our design based on these notions.

Another problem is the software and hardware compatibility issues of the 3D content encryption method. From the practical point of view, it would be very expensive (often impractical) to design a new 3D file format or a new structure of the rendering pipeline for the purpose of 3D content encryption. If the encrypted data does not respect the syntax of most 3D techniques, it cannot be widely adopted for any rendering devices. To this end, our encryption solution is compatible with various 3D file formats, as well as various graphics pipeline structures.

The 3D content encryption method should effectively deform the geometry of 3D content. A potential adversary is able to use 3D pattern recognition techniques to find some relationships among the 3D points, and hence, derive hidden patterns. As a consequence, the question is how
to effectively deform such complex geometric objects so that the underlying information will remain unrecognisable. To address this question, we used non-isometric deformations which annihilate the intrinsic geometry of the shape. The benefit of limiting the design rationale to a class of non-isometric deformations was that it helped to repel any intrinsic geometry similarity between the original and the encrypted 3D content.

3D objects normally have associated data, among which texture images are highly correlated with 3D surface geometry. Therefore, disclosure of texture images can significantly help the potential adversary to reconstruct encrypted 3D objects without knowing the secret key for decryption. To prevent the disclosure of 3D surface geometry from the texture leakage, we proposed a texture encryption scheme which is lightweight and satisfies the security requirement.

3D content encryption is an emerging field, and to the best of our knowledge, there is no rationale for the design and evaluation of secure and usable encryption primitives for 3D content. To this end, we evaluated the security of different encryption primitives, such as permutation-only ciphers. We proved that permutation-only image ciphers are completely broken against chosen-plaintext attacks. We also proposed a precise method to practically obtain the secret key. We expanded the idea of our cryptanalysis to more complex encryption schemes. These cryptanalytic results helped us to better design a secure and lightweight 3D content encryption scheme which is resistant to well-known attacks.

1.4 Organisation of the Thesis

The remainder of this thesis is organised as follows:

- Chapter 1 gives a brief introduction, and outlines the major objectives of this thesis.
- Chapter 2 describes the relevant background of the research topic which contains two parts: the background for 3D content and data security. The critical study of prior works reveals the ‘gap’ in the research undertaken. It also positions the current study in the context of the previous research.
- Chapter 3 identifies the research questions in the context of the security limitations described in Chapter 2. Chapter 3 also defines the hypotheses provided to answer the research questions, and then describes the methodology to validate the hypotheses.
- Chapter 4 proposes a 3D object encryption scheme that is compatible with standard file formats and maintains the semantic requirements of 3D objects, including the dimensional and spatial stability. This chapter is based on the article published in [12].
- Chapter 5 proposes a texture encryption scheme that not only maintains the confidentiality of texture images but also maintains the security of protected 3D models from surface
reconstruction attacks using the data provided by the texture images. This chapter is based on the article published in [13].

- Chapter 6 examines the security of permutation-only image encryption schemes and proves that such ciphers are completely broken against chosen-plaintext attacks. This chapter concludes that no pseudo-random permutations can be realised in order to offer a high level of security against plaintext attacks. This chapter is based on the articles published in [14] and [15].

- Chapter 7 concludes the thesis with some design guidelines for constructing encryption primitives that meet the technical requirements of 3D content applications. This chapter also proposes possible research directions in the field of 3D content encryption.
“Lots of people working in cryptography have no interest, have no deep concern with real application issues. They are not trying to solve a practical engineering problem. They are trying to discover things clever enough to write papers about.”


2

Background and State of the Art in 3D Content Encryption Schemes

2.1 Introduction

3D content encryption is an emerging field of research, the foundations of which, are built on both the data encryption and the multimedia signal processing domains. This chapter critically reviews the background information from these areas, as well as the past research on the topic of 3D content encryption. This chapter describes some of the existing problems and limitations of the current methods. There are five major sections in this chapter:

- Section 2.2 analyses 3D content by illustrating the structure, representation models, format standards, and compression techniques. It also investigates how 3D content is rendered and displayed in virtual environments using the graphics pipeline. This investigation identifies the requirements for maintaining the semantics and renderability of 3D content.

- Section 2.3 explores and analyses data security and details its requirements. 3D content security is a subclass of data security which principally intends to maintain the functionality of 3D content. Therefore, based on 3D content requirements and communication constraints, the required tools for maintaining 3D content security are overviewed.
Section 2.4 explains why a particular encryption primitive is required for encrypting 3D content. This section also explores and analyses the usability issues as well as the technical requirements for maintaining the confidentiality of 3D content.

Section 2.5 analyses the road map to 3D content encryption by reviewing the past research in design of secure pseudo-random number generators (PRNGs), encryption standards, image encryption, and finally, 3D content encryption.

Section 2.6 synthesises the existing problems and limitations found in the literature in relation to 3D content encryption. This helps us identify the research scope and develop a suitable research methodology.

2.2 3D Content

3D content is three-dimensional data which is normally embodied by a sculpture (3D object) and a material in a virtual space. The 3D object demonstrates the sculpture of 3D content and is represented by 3D modelling. Although the 3D object is visualised by its geometry, it is still a far cry from the vision laid out by the concept artist. Hence, it is characterised by a material. The material is represented by a texture map, which precisely defines the physical characteristics of the surface of the object. The texture is essentially an image that is overlaid upon geometry. The structural elements of 3D content are described in following subsections.

2.2.1 Image

An image is a rectangular structure, 2D array, which can be defined by a two-dimensional function, \( f(z(x, y)) \), where \( x \) and \( y \) are spatial coordinates and \( z \) is the colour channel. The amplitude of \( f \) at any pair of coordinates \( (x, y) \) is called the intensity or grey level of the image at the point. The image is named digital if \( f \) and its arguments are all finite and have discrete quantities. Elements of the image array are denoted as pixels. Each pixel is represented by a numerical value which specifies the pixel intensity. The pixel intensity varies within a pre-defined range (bit depth). For instance, in grey value images, the bit depth is 8 bpp (8 bits per pixel). This is due to the capacity of human visual perception to distinguish different levels of intensity. This range varies between 0 and 255 and is demonstrated by a byte.

Colour is defined by using several matrices, one for each colour channel. The most conspicuous example is the RGB representation, where a matrix is allocated for each of the colours red, green, and blue. Another colour representations that takes human perception into account is the YUV model which is represented by luminance and chrominace.
2.2.1 Texture Map

3D models need to be painted; otherwise they would look just like a sculpture. Texture mapping is a fundamental drawing primitive that adds realism to computer graphics by improving surface details. Theoretically, a texture map is just a 2D image that is wrapped onto the geometry of a 3D model to give the illusion of a specified pattern to the complex object [16]. Each element (texel) of the texture map is assigned to a vertex of the 3D object. Depending on the position of the camera, a texel can be bigger or smaller than a pixel. When a polygon is rendered, an interpolating algorithm draws each screen pixel using the surrounding values. Textures can also incorporate surface information (bump map [17]), such as assigning height and depth (roughness) to each pixel, which impacts the way light is received by the surface.

2.2.2 3D Models

A mathematical representation of a 3D object in digital space is called a 3D model. In the literature, there are four main 3D representations for 3D models, which are point cloud, polygonal meshes, surface-based models and volumetric models. In a 3D Cartesian space $\mathbb{R}^3$, where $\mathbb{R}$ is the set of real numbers, each of these representations is defined as follows.

A point cloud is defined as a set of vertices which are represented by $x$, $y$, and $z$ coordinates. These vertices are utilised to demonstrate the external surface of an object. While point clouds can be directly rendered and inspected, they are generally not directly usable in most 3D applications, and therefore are usually converted to polygonal mesh models or surface-based models through a process commonly referred to as surface reconstruction.

A polygonal mesh is defined as a 3-polytope whose elements are its vertices (0-polytopes), edges (1-polytopes) and facets (2-polytopes). The facets may be composed of triangles, quadrilaterals or any other simple convex/concave polygons, which is to simplify the rendering task. This model represents an approximation of the surface. The polygonal meshes are commonly used in most computer graphics applications for the representation of highly complex geometric objects.

A parametric surface is designated as a surface which is specified by a parametric equation with two parameters. Unlike the polygonal representation, this model mathematically represents an exact definition for the surface. This family of 3D surfaces is normally used for computer-aided design (CAD).

A volumetric model is defined as a set of volumetric elements (voxels) which demonstrate a value on a regular grid in a three dimensional space. In this model, the position of each voxel is deduced based upon a comparison between it and the position of other voxels. The volumetric models are commonly used in medical imaging applications, such as magnetic resonance imaging (MRI) and Computed Tomography (CT), in which voxels are utilised to visualise the 3D
Chapter 2. Background and State of the Art in 3D Content Encryption Schemes

2.2.2.1 Creation of 3D Models

3D models are principally created by scanning or modelling. 3D scanners, which are mostly based on lasers or structured light, are becoming very accurate and advanced such that they require less user interaction to capture advanced surface properties [19]. This assists in creating high-quality 3D models from existing objects. There is also a range of computer graphics data. Figure 2.1 illustrates different representation models of the 3D object *Victoria21*. This object is freely available at TOSCA high-resolution 3D object repository [18].

![Figure 2.1: Different representations of Victoria21: (a) a cloud of points, (b) a triangular mesh, (c) a set of parametric surfaces, (d) a set of voxels.](image-url)
software, such as Maya, 3ds Max, and Blender, which content developers can use to enhance scanned models in order to create imaginary objects. The same tools can also be used to assemble 3D objects into complex forms and scenes, to design animations.

![Diagram of 3D objects](image)

**Figure 2.2:** Sources of 3D objects.

### 2.2.3 Compression

3D objects are complex shapes broadly used in many graphical applications. Maintaining such shapes in raw data format consumes a large amount of space. The need for efficient tools to reduce the size of 3D content becomes even more significant when considering the transmission latency of low-bandwidth networks. This is precisely the objective of compression techniques, that is maintaining a minimal bit rate while preserving high visual quality. A compression algorithm normally includes the following two parts:

- A succinct/synthetic representation of 3D content; and
- Coding techniques, such as quantisation, prediction and entropic encoding, to suppress redundancy.

During the last 30 years, a wide community of researchers has worked on compression methods. The most well-known of these, the joint photographic experts group (JPEG), is largely the result of efforts of this part of the scientific community since 1987. JPEG is the current standard for the compression of 2D images and transmission of digital television over a reduced bandwidth asymmetric digital subscriber line (ADSL).

The main objective of a compression technique is to find an optimal compromise between the bit rate and the distortion. Compression methods divide into lossless and lossy. The former perfectly reconstructs the original data with limited compression ratios while the latter controls a loss of information to enhance the bit rate in low-bandwidth networks. The loss of data normally appears in the form of mosaic effects, reduced colour quality, slight local deformations, or an imprecise reconstruction of a 3D object. A compression algorithm is generally composed of three parts: redundancy reduction, quantisation, and coding.

Redundancy is defined as the amount of similarity between the symbols generated by an information source. It may be spatial (in neighbouring pixels or vertices), spectral (between the
red, green and blue components, for instance), or temporal (between successive planes of animated 3D objects). Compression techniques detect the presence of different types of redundancy using advanced tools, such as predictions and transformations like wavelets. These tools transform the information into a compact and uncorrelated representation. Uncorrelated information may be represented by integer, real or complex values in a given dynamic range. These formats and ranges are often incompatible with the average number of bits per symbol associated to the channel capacity. In such cases, quantisation methods have to be used. Quantisation methods are either applied to scalar values like colour intensities and vertex coordinates, or applied to vectors like blocks of neighbouring vertices. The quantisation is a lossless process which represents the values by a binary code of fixed length. In the coding stage, the average number of bits allocated to each quantised value is reduced.

The majority of 3D compression methods are applied to mesh representation. This is due to the fact that it is the most used representation in computer graphics and geometric modelling, and it is quite large in terms of the amount of data. For a discussion of mesh compression, refer to [20]. Other existing 3D representations, such as point clouds [21], parametric surfaces [22] and voxels [23], have also been considered by the 3D compression community.

2.2.4 3D Content Formats

To permit the content publication or distribution between applications and the network, a number of standards (140 file formats) have been developed for representation of 3D content [24]. These standards employ different methods to describe and encode content, making them more appropriate either for content exchange or content broadcast. In this subsection, selected standards suitable for content publishing, namely virtual reality modelling language (VRML) [25], extensible 3D (X3D) [26], MPEG-4 [27], universal 3D (U3D) [28], and collaborative design activity (COLLADA) [29], which are platform-independent, and OBJ [30] are presented. These standards have been designed for different purposes and therefore have different capabilities in handling different types of media content such as sounds, movies, and graphics. Some of them are mainly used for content exchange, while others are more suitable for publishing. Table 2.1 summarises the main characteristics of the above-mentioned 3D content standards.

VRML [25], introduced in 1995, is ISO certified and the most widely supported 3D format for tools and viewers. VRML is a text file format which supports 3D geometry, animation, and scripting. This format does not have support for 2D graphics but it is capable of demonstrating both static and animated 3D objects, linked with other media, such as text, sounds, movies, and images. VRML files, often used interchangeably with the term worlds, are in plain text format. One disadvantage of VRML is the lack of compression support. Thus, VRML files are usually compressed by gzip for quick transmission over the internet. Streaming of graphical content is also not allowed by VRML, making it difficult to update the scene graph in real time applications.
X3D [26] is a royalty-free open standard for the representation of 3D computer graphics. X3D is an XML-based ISO/IEC standard, which provides a system for the storage, retrieval and playback of real time 3D graphics by specifying a declarative geometry language, a runtime engine, and an application programming interface (API). X3D has been designed to keep backward compatibility with its predecessor, VRML; thus, it includes both XML and VRML syntaxes. In addition, it can also be delivered in compressed format. X3D supports streaming of video and audio, but not 3D content, which makes it inappropriate for server types of 3D multimedia applications. Despite advantages of X3D over VRML, it is still not accepted by the industry, which limits its usage to research.

MPEG-4 [27] is an ISO/IEC standard developed by the moving picture experts group (MPEG). This standard offers a comprehensive set of tools for transmission and compression of different kinds of multimedia content, including audio, video and VRML objects. The MPEG-4 standard enables grouping of primitive media objects, such as still images, video objects, audio objects, and text and graphics, to construct complex media objects. In addition, it allows applying transformations to media objects to change their geometrical attributes, such as the position and appearance. The MPEG-4 standard also makes it possible to change the user’s viewing and listening points in the scene. The scene representation in MPEG-4 standard has grown out of VRML/X3D and has been extended to cover full MPEG-4 functionality [31].

U3D [28] is a format standard established by a number of 3D Industry companies such as Intel, Hewlett-Packard, Adobe Systems. This format, which was designed to enable presentation of 3D CAD models, was standardised by ECMA International as ECMA-363 [28]. The prominent features of the U3D file format involve the capability of runtime modification of geometry, domain-specific compression, continuous level of detail [32], progressive data streaming and playback, free-form surfaces, key-frame and bones-based animation, and extensibility. The
U3D format is also supported by portable document format (PDF); thus, 3D objects in U3D format can be embedded into PDF documents and interactively visualised by Acrobat Reader.

COLLADA [29] is an intermediary format designed by Sony Computer Entertainment for content transaction between interactive 3D applications. In 2005, this format was accepted as an industry standard by the Khronos Group, a member-funded industry consortium, which now shares the copyright with Sony [29]. Likewise X3D, COLLADA is a royalty-free open standard [33]. COLLADA is based on XML and targets mostly the game industry. This format can be used to specify how the data is displayed on different terminals through the specification of different profiles, therefore allowing the artist to tweak the data specifically. Although COLLADA is not intended to be a content publishing format, there are browsers capable of displaying COLLADA content. In addition to the strong support for 2D and 3D graphical primitives, other types of media like video or sound and are not supported. This is a big disadvantage since it means that COLLADA by itself cannot be used for storing data needed for more complex multimedia applications and scenes. An additional disadvantage is that there is no support for user interactivity or scripts, therefore limiting its usage only for storing data. Furthermore, there is no support for compression and no support for streaming of data.

The OBJ [30] standard is a text based, open file format developed by Wavefront Technologies, which can be used to store and exchange 3D data. The file format represents only 3D geometry. In other words, it includes the position of each vertex, the position of each texture coordinate vertex, normals, the polygonal faces defined as a list of vertices, and texture vertices [34]. This file format has been adopted by many 3D graphics application vendors; hence, it is a universally accepted format.

2.2.5 Virtual Worlds

The major factor that differentiates 3D content from a binary stream is data visualisation. 3D content is normally visualised in a virtual scene which is a part of a virtual world. Virtual worlds are multi-user synthetic environments that enable users to interact with each other through 3D graphical environments hosted on networked computers [35]. Virtual worlds are gradually becoming an effective medium of communication for social and commercial interactions. They also share many similarities with existing telecommunication technologies such as e-commerce and entertainment. The content and activities within virtual worlds may be a model of real life, or be totally imaginary. Hence, such worlds are sometimes referred to as alternative realities or metaverses. When in-world events are synchronised with the events in the real world, they are termed mixed reality events.

Due to widespread applicability of virtual worlds, business and commerce companies have substantial interest in transplanting existing real world business models into virtual worlds, and
target virtual world users using the same methods as in the real world. However, due to the technical weaknesses of virtual world platforms, several security threats may occur. For instance, servers that store sensitive user information may undergo severe intrusions from malicious entities. In virtual reality, every user has a particular camera to view the virtual world. So, they can interact closely with the 3D graphics. Hence, the confidentiality of privacy sensitive information in the public virtual worlds needs to be preserved.

2.2.6 Rendering Pipeline

In computer architecture, a pipeline refers to a sequence of data processing elements, where the output of one element is used as the input of the next one. In the rendering process, a representation of a 3D object, which is mainly composed of polygons, is passed through the graphics driver software on the way to the graphics processing unit (GPU). The GPU accelerates the memory-intensive work of texture mapping and rendering polygons by speeding up geometric calculations, such as lightening and translation of vertices into different coordinate systems. The GPU then maps the scene geometry to pixels (rasterisation process) and builds the output in the frame buffer to display the visual data. Figure 2.3 shows the abstracted graphics pipeline.

![Graphic Pipeline Diagram](image)

**Figure 2.3:** Graphics pipeline.

2.2.7 Virtual Camera

The virtual camera is a window to view the virtual 3D world, which exposes different perspectives of 3D objects by viewing 3D scenes from different angles. In virtual worlds, each user is incorporated with a virtual camera to view the virtual world. The field of view is the range of the virtual 3D world that is observed by the camera at any given instant. The area captured by the camera, which is a part of the field of view, is called the camera view plane. In other words, the camera view plane is the frame through which users can see the virtual world. The visible 3D
scene in the field of view is projected into the camera view plane. In technical term, the visible content in the camera view plane is the final product of the rendering pipeline. Figure 2.4 shows a camera view plane of a 3D scene.

![Camera view plane](image)

**Figure 2.4:** Camera view plane.

### 2.3 Data Security

Throughout history, from rock and cave paintings all the way to Hieroglyphics, and to modernism and today’s computer era, human beings have been looking for ways to protect themselves from unwelcome intrusion. Such intrusion may arise from both accidental and intentional events. Protection against accidental events is addressed by safety practices while protection against intentional events is addressed by security and privacy practices. Security provides the ability to be confident about the information being respected, while privacy intersects with security and provides the ability to decide what personal information goes where. Psychological studies acknowledge that safety and security requirements, after physiological needs, are the most primitive and fundamental requirements of human beings (Figure 2.5) [36]. In addition to historical and physiological grounds, security is an increasing concern in the current multimedia defined world.

Regardless of the data type, security is principally defined as the practice of protecting data from unauthorised access, use, alteration, disclosure, disruption, perusal, scrutiny, recording or annihilation [37]. Different aspects of security are stated as follows:

- *Availability* ensures that authorised parties are able to access the information when needed;
• **Authentication** ensures the senders of messages are who they claim to be;

• **Authorisation** ensures whether a requester has the permission to receive a service or perform an operation. Access control, which is the selective restriction of access to a resource, is an example of authorisation;

• **Confidentiality** ensures secure communication by keeping the information secret from all except those who are authorised;

• **Integrity** ensures that the data transmission has not been tampered with by an unauthorised user;

• **Non-repudiation** ensures that transmitted data has been sent and received by the parties claiming to have sent and received the data. It also verifies who the author of the data is; and

• **Copyright protection** ensures the ownership of data.

Generally, these aspects can be addressed by a number of technical solutions such as encryption and digital watermarking. Encryption and watermarking are two different techniques used for different purposes [7]. In encryption, only authorised parties holding decryption keys can access content in clear text. Encryption provides no protection once the content is decrypted. Decrypted content can be easily altered and copied by an untrustworthy user without the permission of the content owner. Therefore, to complement encryption and detect the content piracy, watermarking is applied which embeds the owner’s private information, namely a watermark,
into the original content and extracts it from a questionable content when the ownership needs to be resolved [7]. As shown in Figure 2.6, the focus of this research is encryption.

2.3.1 3D Content Security

In the context of information creation, processing, transmission and storage, 3D content implies on a higher level representation or semantics compared to 1D and 2D data. Therefore, characteristics of 3D content are significantly different from other types of data. For instance, the data structure, data size, representation, required bitrate, compression complexity, and security requirements may deviate considerably which introduces several challenges in terms of 3D content protection.

3D content security is a subclass of data security which principally intends to maintain the functionality of 3D content. This field of security has a challenging interdisciplinary nature and faces many issues including the requirements of multimedia communication, multimedia retrieval, multimedia compression and hardware resource usage. Due to the nature of 3D content, traditional forms of data security cannot sufficiently address the needs of 3D content security because they mainly process information at the bit-level which is oblivious to the semantics of the information.

Similar to data security, 3D content security involves several different requirements including authentication, confidentiality, access control, and copyright protection. Generally, content confidentiality, authentication and access control are addressed by encryption, through which only authorised parties holding decryption keys can access content in clear text. On the other hand, the copyright protection is normally addressed by digital watermarking. In this thesis, we are only concerned with providing confidentiality using encryption. Figure 2.6 depicts the tools used to satisfy the data confidentiality requirement and the green area shows our research interest.

2.3.2 Cryptology

Cryptology is the practice and study of mathematical techniques to provide secure communication in the digital world. A comprehensive overview of cryptology can be found in the Handbook of Applied Cryptography by Menezes et al. [38]. Cryptology embraces both cryptography and cryptanalysis. Cryptography is the art of designing efficient encryption/decryption systems while cryptanalysis is the art of determining the security weaknesses of cryptographic algorithms and examines whether they are vulnerable to certain attacks.
Chapter 2. Background and State of the Art in 3D Content Encryption Schemes

2.3.2.1 Cryptography

The aim of cryptography is to authenticate communicating entities, ensure data privacy, preserve data integrity, and prevent repudiation in a private communication. The problem of private communication is often featured as a transaction between two end points: the sender and receiver. This setting is described in Figure 2.7.

To provide a confidential communication, an encryption/decryption scheme (cryptosystem) is used to convert the plaintext message into the ciphertext using the key. The objective of a cryptosystem is to enable two entities, who have already shared a key through a secure channel, to securely communicate over an insecure channel. A cryptosystem, which is often used interchangeably with cipher, is formally defined as follows:

Definition 2.1. A cryptosystem (cipher) is a five tuple \((\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})\), where \(\mathcal{P}\) is the plaintext space, \(\mathcal{C}\) is the ciphertext space, \(\mathcal{K}\) is the key space. For each key \(k \in \mathcal{K}\), there is an encryption
For every key $k \in K$, the function pair $E_k: \mathcal{P} \rightarrow \mathcal{C}$ and $D_k: \mathcal{C} \rightarrow \mathcal{P}$ satisfy $D_k(E_k(p)) = p$ for every plaintext $p \in \mathcal{P}$.

Cryptographic designs are synthesised based upon cryptographic primitives, including pseudo-random functions, one-way functions, and low-level functions. Cryptographic primitives are well-established, low-level basic building blocks of cryptographic algorithms, which are designed for specific purposes with the aim of accomplishing a number of security goals.

Depending on whether encryption and decryption use the same key or not, a cryptosystem can be either symmetric or asymmetric. In the former, a single key is used for the encryption and decryption, while in the latter, two keys are used; one key is visible to the initiator of the encryption process and is useful for signing and encrypting documents; the other key is publicly available and is used for verifying the author of signatures and also for decrypting associated documents.

Encryption methods can be applied in both storage and transmission applications. In multimedia applications, due to the large amount of data involved in multimedia content, a huge storage capacity or transmission bandwidth is required. To provide a reasonable execution performance for encrypting such bulky data, only symmetric encryption, as opposed to public key cryptography, can be used. With respect to this, public key techniques are normally used for key exchange or signature generation only.

Block ciphers and stream ciphers are two classes of symmetric encryption schemes. The former encrypts the group of characters of the plaintext at one time, while the latter is stateless and encrypts the individual characters of the plaintext one by one. Stream ciphers are not only appropriate for the resource constrained devices but also suitable for time critical applications which require performance extremes in speed, memory, and power consumption.

As stated above, a symmetric cipher is used to protect plaintext from undesirable disclosures. However, an adversary, who may have access to everything transmitted through the insecure channel, aims to “break” the cipher by recovering the plaintexts from intercepted ciphertexts. A cipher is completely broken if the adversary can determine the complete secret key, and it is partially broken if the plaintext is partially recovered from intercepted ciphertext, but the secret key is not disclosed. Security is always relative to threats. Security has different notions, which are explained below:

- **Unconditional and Computational Security:**

  The theory of secrecy was firstly established by Claude Shannon in [39], where he introduced two notions of security: unconditional security and computational security.

  - **Unconditional security:** In an unconditionally (perfectly) secure cipher, the ciphertext provides no information about the plaintext. In other words, a potential adversary with unlimited computational power must not be able to recover any part of the
plaintext by comparing any plaintext/ciphertext pair. In probability theory, this is described as
\[ Pr\{P = p \mid C = c\} = Pr\{P = p\}, \]
for any \( p \in P \) and \( c \in C \), where \( P \) and \( C \) denote the random variables corresponding to the plaintext and ciphertext. However, from the information theory point of view, Shannon showed that unconditional secrecy infers
\[ H(K) \geq H(P), \]
where \( H(\cdot) \) is the Shannon entropy. This notion of perfect secrecy implies that (1) the key length must be at least as large as the plaintext length, and (2) the key must be a truly random sequence. Thus, it is not easy to build an unconditionally secure cipher in practice. However, unconditional secrecy is attainable by Vernam cipher (one time pad) which XORs the plaintext with a key of the same length to generate the ciphertext. Unfortunately, due to the mentioned difficulties, applicability of Vernam cipher is very limited.

- **Computational security**: According to Kerckhoffs’s cryptographic principles, a cipher must be practically, if not mathematically, indecipherable. However, due to the inherent problem of random key generation with the size of plaintext, designing a perfectly secure cipher is difficult. As cryptanalysts in practice have limited computational power, it is possible to provide an alternative solution (computational security) by encrypting the plaintext using a pseudo-random key shorter than the plaintext length. A cryptographic primitive is said to be computationally secure if the best algorithm for breaking it requires at least \( 2^n \) operations, where \( n \) is some large fixed number. In this case, it is said that the cryptosystem provides \( n \) bits of security.

- **Provable and Heuristic Security**:

Security also includes other notions, such as provable security and heuristic security.

- **Provable security**: This notion of security refers to any type or level of security that can be proven. A cryptographic primitive has provable security if its security can be reduced to a known, well-studied problem that is believed to be hard. If the cryptosystem can be broken in some specific way, then it would be possible to efficiently solve the reduced problem which is believed to be difficult. Problems closely related to integer factorisation (RSA) and discrete logarithm (ElGamal) are such widely-accepted examples. In this model of security, security is preserved against efficient adversaries who have access to the system as well as enough computational resources.

- **Heuristic security**: As opposed to provable security, a cryptographic primitive has a heuristic security if there are no currently known cryptanalytic techniques to break it, or its security cannot be proven in any sense.
2.3.2.2 Cryptanalysis

Designing a cryptographic algorithm is not an easy task and requires a number of consecutive stages. As well as the basics of cryptography, different well-known cryptanalytic methods need to be studied. As a basic starting point, according to Kerckhoff’s principle [40], security of an algorithm cannot rely on its secrecy and it must rely on the secret parameter (secret key).

Based on the amount of information available to the attacker, there are different cryptanalytic attacks, each defined based on a different scenario to obtain the secret key. These attacks are numbered as follows [43]:

- **Ciphertext-only**: the cryptanalyst has access only to a string of ciphertext.
- **Known-plaintext**: the cryptanalyst has a string of plaintext and the corresponding ciphertext.
- **Chosen-plaintext**: the cryptanalyst has obtained temporary access to the encryption machinery which can construct the ciphertexts corresponding to an arbitrary set of plaintexts.
- **Chosen-ciphertext**: the cryptanalyst has obtained temporary access to the decryption machinery and so is able to construct the plaintexts corresponding to an arbitrary set of ciphertexts.
- **Chosen-key**: the cryptanalyst can partially control the secret key. This allows the adversary to choose key transformations and consequently alter the key and then request encryptions under the modified keys.

From the cryptographic point of view, a strong encryption scheme should resist all of the above-mentioned attacks. In other words, a cipher should be thoroughly analysed before being used in practice.

2.3.2.3 Statistical Analysis

In the literature, there are some statistical metrics which can be used to measure the cipher’s resistance to some typical attacks, such as key space analysis, key sensitivity analysis, plaintext sensitivity analysis, randomness testing, and performance analysis.

- **Key Space Analysis**: Theoretically, the security level of a cryptosystem is dependent on its key length. In literature, there are various recommendations for the appropriate key length of a particular encryption system [44], [45], [46]. According to the guidelines released by the national institute for standards and technology (NIST) [46], 128 bit key length is an acceptable margin for designing secure symmetric-key algorithms until 2030.
• **Plaintext Sensitivity Analysis**: In general, a desirable property for a ciphertext is being sensitive to minor alternations in the plaintext. From the security point of view, if changes to one element do not spread to neighbouring elements, the cipher would fall to a chosen-plaintext type of attack. To study the relationship between the plaintext and ciphertext, the adversary slightly changes a single element of the plaintext, for instance, the single bit of a pixel value or the position of one point in the point cloud, and observes the changes in the ciphertext. By this method, the meaningful relationship between the plaintext/ciphertext pair can be found, that further facilitates in determining the secret key. If a small change in a single element of the plaintext affects the entire ciphertext, then the differential attack becomes practically infeasible.

• **Key Sensitivity Analysis**: Similar to plaintext sensitivity, a good encryption scheme should be sensitive to changes to the secret key. In other words, a change in a single bit of the secret key should produce a completely different ciphertext. Being sensitive to the changes of the secret key ensures that the cipher image cannot be decrypted correctly even if there is only a small difference between the encryption and decryption keys.

• **Correlation Analysis**: In the plain data, each element is highly correlated with its adjacent element. An ideal encryption algorithm should produce a cipher data with no such correlation in the adjacent elements. A high correlation among the elements of the cipher data helps the adversary identify the underlying pattern of the original data. Hence, such cipher is vulnerable to correlation attacks.

• **Randomness Test**: To ensure the security of a cryptosystem, the cipher must have some probabilistic properties such as good distribution, long period, high complexity and efficiency. In particular, the outputs of a cryptosystem must be unpredictable in the absence of knowledge of the inputs. There are an infinite number of statistical tests, such as the DIEHARD battery of tests [47], the FIPS 140-2 tests [48], and the NIST Special Publication 800-22 test suit [49], each assessing the presence or absence of a pattern which, if detected, would indicate that the sequence is non-random. The DIEHARD tests are a battery of statistical tests which were developed by George Marsaglia in 1995. These tests include birthday spacing, overlapping permutations, rank of matrices, monkey tests, count the ones, parking lot test, minimum distance test, random spheres test, squeeze test, overlapping sums test, runs test, and the craps test. The mathematical description of each test can be found at [47]. The FIPS 140-2 tests are the statistical tests developed by NIST in for evaluating encryptions and random number generators. They include four statistical tests of randomness: the frequency (monobit) test, the poker test, the runs test, and the long runs test. Details of these tests can be found at [48]. Later, NIST proposed an updated set of standard statistical tests including a frequency (monobits) test, a frequency test within a block, a runs test, a test for the longest run of ones in a block, a binary matrix
rank test, a discrete Fourier transform (spectral) test, a non-overlapping template matching test, an overlapping template matching test, Maurer’s “universal statistical” test, a linear complexity test, a serial test, an approximate entropy test, a cumulative sums (Cusum) test, a random excursions test, and a random excursions variant test, to assess the randomness of binary sequences produced by either hardware or software based cryptographic random or pseudo-random number generators. These tests focus on a variety of different types of non-randomness that could exist in a sequence. The mathematical description of each test can be found at [49]. The NIST framework, like many statistical tests, is based on hypothesis testing. A hypothesis test is a procedure for determining if an assertion about a characteristic of a population is reasonable. In all tests if the computed $P$-value is $< 0.01$, then it is concluded that the sequence is non-random; otherwise, it is concluded that the sequence is random.

2.3.2.4 Performance Analysis

In addition to statistical analysis for security evaluation, the encryption performance is also an important factor to consider, especially for real-time applications which require a high level of efficiency. Generally, encryption performance hinges upon the structure of the central processing unit, programming language, memory size and the operating system. Performance of a cipher is normally measured by calculating the following measures:

- **Encryption time**: The encryption time is considered as the processing time that an encryption algorithm takes to convert a plaintext into a ciphertext.

- **Encryption speed in clocks per byte**: To allow applicable comparisons, encryption speed is calculated in clocks per byte which is the number of required processor clock cycles to process each byte. The number of central processing unit (CPU) clock cycles is a metric reflecting the amount of energy consumption of the CPU while operating on encryption operations.

- **Computational complexity**: The computational complexity estimates the number of elementary operations executed by an algorithm, where each elementary operation consumes a fixed amount of time to accomplish a task.

- **Space complexity**: The space complexity estimates the number of memory cells that an algorithm requires to perform a specific task.

2.4 3D Content Encryption: Why?

Since the 1970s, a large number of symmetric ciphers have been proposed, among which some have been standardised and widely adopted all over the world, that is, DES [50] and AES [51].
So, it seems natural to use established and tested ciphers to encrypt 3D content bit by bit. This simple and naive approach has already been used in many digital rights management (DRM) systems [52], [53]. For instance, in secure real-time transport protocol (SRTP) [54], data is packetised and each packet is independently encrypted using a conventional cipher, such as AES. This approach provides the same level of security as that of the conventionally used cipher. However, due to some special features, naive encryption may not be a suitable solution for many 3D applications. The problem of 3D content encryption is beyond the application of established and well-known encryption algorithms. This is primarily due to the structure of 3D content and the way it is used commercially. Unlike data encryption, where a complete bitstream is encrypted, 3D content encryption introduces several challenges.

Firstly, 3D content is further than a binary stream and is a composition of a 3D object and a texture image wrapped around it. It may be of large volume and high redundancy. Real-time interactions with such big data, such as viewing, editing, scaling, searching and watermark embedding, are costly operations. Hence, it is necessary to decrease the computational cost of encryption and decryption processes as much as possible to allow other possible operations, while safeguarding the smooth rendering. In addition, in many 3D virtual worlds, such as Second Life [9] and World of Warcraft [55], where users can socialise, connect and create 3D contents, there are always a large number of 3D objects stored in many servers, and a large number of 3D content streams are transmitted from these servers to end users. In such cases, the encryption/decryption process places a severe load on the servers which may be too high to ensure smooth running of the services. On the other hand, each part of 3D content has a different structure and hence has different requirements, which need to be considered for encryption purposes. Due to the shape and geometric layout of 3D objects, they contain a large amount of valuable information. However, 3D content protection by solely 3D objects encryption is not a complete solution. In addition to 3D objects, texture images contain intelligible information which is due to the strong correlation among adjacent texels. As each texel is assigned to a particular vertex, texture patterns provide strong cues to the surface orientation, curvature and 3D surface geometry. Accordingly, texture leakage may lead to a disclosure of the 3D surface geometry. As a result, confidentiality of texture images, as well as 3D objects, needs to be preserved.

Despite the structural and application-based concerns, it can still be argued that with the recent advances in parallel processing and GPU processing capabilities, real-time naive encryption of 3D content is achievable. However, it should be noted that bulk encryption of 3D content basically annihilates the format compliance, and hence, the encrypted content cannot be accessed. More importantly, rendering engines are unaware of the encryption process; they cannot parse the non-compliant encrypted content in the virtual environment and thus show nothing in the place of the object. Essentially, parsing such content causes a critical crash in the rendering process. To evade this problem, it is possible to hide the content or display nothing, without undergoing the encryption process. This is against one of Kerckhoffs’s cryptographic principles.
Obscurity cannot guarantee security. An alternative to this may be to encrypt the content but not render it. This may not be an ideal solution. In some applications, such as in virtual museums and 3D e-commerce, if users have no means of noticing that something is missing, it is then unlikely that they will find that something is missing. In practical terms, the motivation to pay for having access to 3D content would be lacking.

A solution to these problems is to use a format compliant encryption which can preserve the structure of 3D content and independently encrypt its structural elements. Format compliant encryption enables the post processing of multimedia data in the encrypted domain, such as editing, scaling, searching and watermark embedding, without decryption. This solution is already applied in Extensible Markup Language (XML) encryption [56]. XML encryption considers contents of an XML element as a binary stream and thus, while keeping the syntax format, it can encrypt any kind of arbitrary data using conventional ciphers. This approach also offers granularity. In other words, users can choose to encrypt only a few of many objects in a complicated 3D scene. However, this solution is not complete. Firstly, unlike binary streams, 3D content has a 3D geometry and in many applications, such as virtual museums and 3D e-commerce, it is required to display such geometry in a 3D space. 3D objects should be confined to the virtual space in order to be displayed. If such consideration is not taken into account, then encrypted objects may exceed the viewing screen resolution or they may overlap with other objects of the virtual world. Hence, the encryption outcome may conceal other objects behind its surface. This decreases users’ observation capability. Conventional ciphers, such as AES [51], are oblivious to dimensional and spatial stability and hence, the rendering may spill out their encrypted outcome and corrupt the whole 3D scene. In practical terms, conventional ciphers destroy the spatial and dimensional stability.

An example for the dimensional and spatial requirement is the case of virtual 3D simulators, where users can practice many real world experiments by using virtual objects. Users can change their viewpoint or rotate and move objects for better viewing. In such applications, 3D simulation companies may provide access to the full scene only to paid customers. Despite such marketing strategies, unpaid customers may still need to interact and experiment with unprotected 3D objects and be able to modify the geometries. Such modifications should not affect the existing graphical elements that would be indiscernible due to the protection framework. Another example is the game production workflow where a crew of level designers, artists and programmers cooperate under the supervision of a producer. The output of the production workflow is a game program and a series of game assets to be consumed by the game program. These assets include 3D models and images data. The 3D designs are normally updated during the production workflow. For instance, some parts or textures can be added to or added from a particular object. In the production workflow, the design studio may not be willing to give access to the full scene to a particular crew. However, this crew may need to have access to visual
cues in order to insert graphical elements without striking existing ones that would be invisible because of the protection framework.

In general, the encryption and decryption procedures are computationally demanding and time consuming, and they are computationally expensive for power constrained real time communication. Thus, encrypting the entire bitstream may not be a suitable solution for high bitrate multimedia, particularly when the transmission is done over a low bitrate communication channel. In addition, reduction in the computational cost of the encryption process would allow more processes to be executed in parallel. A solution to this is a light weight encryption process. While this approach may instigate security threats, it has higher efficiency. On the other hand, in many commercial applications such as try-and-buy online services, users need to have a preview of the encrypted content to check whether it takes their interest, and only the authorised or paid consumers can obtain the original content with high quality. A solution to this is selective encryption. A degraded preview supports many operations such as search and aggregation on encrypted objects while maintaining a certain degree of privacy, can also be used as a copyright protection method, because only the owner is capable of revealing the original quality of 3D content. A solution to this is to encrypt a subset of the bitstream. Although some content may leak after encryption, it has several benefits. Firstly, while attaining an acceptable level of security, the data size (or required memory) and therefore the computational overhead of encryption, is reduced. In addition, a degraded version of visual information not only can motivate users to pay for the full version, but also has a lower cost compared to the full encryption.

2.4.1 Technical Requirements

3D content has a number of requirements. It should be noted that some requirements are consistent with each other, while other requirements may conflict. Different applications may have different specific requirements with a different order of priorities. Therefore, in the design of a practical 3D content cryptosystem, it is always necessary to make a careful compromise between conflicting requirements. The important requirements of 3D content are described as follows.

*Format Compliance:* 3D content is organised and stored in several differing formats. File formats are designated for different purposes, such as content exchange or content broadcast, and therefore have different capabilities in handling different types of media. In addition to the file format, compression and coding algorithms normally append a data stream with format information, such as a file header and a file tail, to 3D content data. This information is employed by decoders to reconstruct the data successfully. If the format is modified by the encryption then the encrypted content cannot be accessed for simple operations, such as coding/decoding, displaying and editing. In such cases, the encryption is incompatible with the compression operation. Encrypting the 3D content without effecting the format information preserves the format of the encrypted data. Format-compliant (syntax-aware [57]) encryption increases the
usability of the 3D content cryptosystem because it allows the encrypted content to be accessed, controlled and edited, without decryption. Operating directly on the encrypted data without decryption reduces the complexity cost.

Granularity: 3D content database can contain multiple scenes and each scene can contain multiple objects. In addition, objects can contain multiple materials which can be comprised of many textures. Encrypting the entire 3D content database (transparent encryption) may not be a suitable solution because it imposes a high computational load. Besides, it may not be necessary to encrypt the entire database and only privacy-sensitive contents may require protection. Hence, a subtle way of protecting 3D content is to only encrypt chosen elements. This is possible because 3D content database is often made up of a number of distinct textured 3D objects positioned in a setting. When each 3D object is coded as a separate entity, it becomes possible to protect each 3D object individually and it is not necessary to protect all of them.

Semantic Constraints: In realistic 3D scenes, there are often thousands of 3D objects which interact with each other by a 3D geometric shape with a recognisable boundary. This imposes a number of crucial semantic constraints that arise from the geometric configuration of the objects. 3D objects are solids and cannot intrude into one another. In practical terms, the intersection of any pair of 3D objects is either an empty set or at most of a surface. Hence, to display 3D objects with no visual inconsistencies, semantic constraints need to be satisfied. To this end, it is necessary to preserve dimensional and spatial stability.

- Dimensional stability: The dimensions of 3D objects must be less than or equal to the dimensions of the virtual world. In addition, no part of the dimensions of 3D objects must be allowed to fall outside the virtual world.

- Spatial stability: To prevent interference among objects, each object should be placed in its own designated location. The location of 3D objects must be within the virtual world. In addition, the location of 3D objects must be such that no part of it falls outside the virtual world.

Object Representation: In comparison with text, sound, 2D images and videos, 3D objects are more complex because they have many different representations. A 2D image is uniquely represented by a 2D matrix whose elements (pixels) depict colour intensity. All devices and systems that produce 2D images render this same representation. However, 3D objects have different kinds of representations depending on the type of 3D scanning device used. For instance, 3D laser scanners produce a cloud of points while CT scanners create a volumetric model in the form of a 3D cube. In addition, different 3D applications, such as medical applications and CAD, use different representations. Each representation employs different primitives with different data semantics to illustrate 3D objects. This suggests different encryption schemes for each representation.
**File Format:** 3D content is encapsulated as a database within a file format. A file format describes the way 3D content information is organised in a computer file. There are many different file formats available. Each format employs different primitives to describe and encode 3D content, making it more appropriate either for content exchange or content broadcast. The choice of file format is normally based upon many considerations, such as file size, loading speed, compatibility, and readability. Accordingly, it is important to consider the format primitives when designing a new cipher.

**Real-time Performance:** 3D content is normally very large but the communication bandwidth is usually limited; therefore, real-time streaming of 3D content is of great importance to content providers. This is mainly because users will not wait a long time to communicate with secure 3D content. Hence, 3D content encryption/decryption schemes should not substantially increase the latent transmission time. However, 3D content normally has a large size and encrypting such bulky data induces many performance overheads, which can be too expensive for real-time applications. Hence, to meet the real-time requirements, appropriate encryption and decryption schemes, which bring reasonable delay, need to be selected.

**Complexity:** Many 3D applications require real-time rendering of 3D content. Due to the processing of a bulky data stream, 3D content encryption and decryption techniques may incur significant processing overheads. This can alleviate smooth rendering and limit usability of other real-time functions. Hence, it is necessary to keep encryption complexity as low as possible to save computational resources, battery power, time and bandwidth.

**Compression Efficiency:** 3D content is normally compressed to reduce the required storage space, transmission time and bandwidth. Compression of a completely random content has a very poor performance, and there is a possibility of it even growing in size, because there is no statistical redundancy to eliminate. This makes encrypted content a poor candidate for compression. On the other hand, a completely predictable content can generally be compressed very well. However, most secure encryption mechanisms generate a pseudo-random (not a truly random) ciphertext. Also, Johnson et al. [58] showed that in the case of an ideal Gaussian source, compressing the encrypted data by the theory of distributed source coding gives the same compression gain as compressing the unencrypted data. Hence, depending on the application, encryption may be applied to 3D content before, during or after compression. However, in all cases, encryption should not change the compression ratio or should at least keep the changes in a small range. In other words, 3D content encryption should not introduce many computational overheads. This is particularly important in limited bandwidth applications, such as peer-to-peer (P2P) 3D streaming of the virtual environment content.

**Level of Security:** As varied applications have dissimilar requirements, the acceptable level of security is different. For instance, military applications require a stress on complete confidentiality, however, for pay per view applications, perceptual degradation rather than high end security is sufficient. To identify an optimal security level, the cost of protection should be
compared with the value of the 3D content information which needs to be protected. If the content is not of great value, then relatively light-weight encryption may be sufficient. On the other hand, if the 3D content is of great value, the cryptographic security level must be the highest possible. However, maintaining a high level of security in real-time communication with limited-bandwidth is a challenging task. Generally, if the cost of attack is higher than the monetary value of content, or the time complexity of the successful cryptanalysis is longer than the transmission time, then the encryption algorithm may be considered as secure. For instance, in many real-world applications which have a high content rate, such as public broadcasting, the information may not be of great value or may lose its importance over time of transmission. In contrast, there are some applications, such as 3D e-commerce, which require a higher level of confidentiality, regardless of real-time considerations. With respect to real-time and limited-bandwidth limitations, maintaining a high level of security is a challenging task.

**Perceptibility (Visual Degradation):** Based on the application, 3D content may undergo a lenient or strict encryption process. Hence, some content may leak after encryption. Although, this may instigate security threats, it has several benefits. A degraded version of visual information not only motivates users to pay for the full version, but also has a lower cost compared to the full encryption. It is necessary to measure the perceptual distortion of the encrypted content with respect to the original content. The level of visual degradation is normally adjusted with notice to the target application and requirements. The metrics used in the literature to measure visual degradation are the PSNR [59], encryption quality [60] and scrambling degree [61]. Visual degradation is a subjective criterion which is why it is difficult to define a threshold for acceptable visual distortion regarding a given application.

### 2.5 Road Map to 3D Content Encryption

Undoubtedly, visual communication is the most operative form of communication, which always precedes other forms. The history of visual communication starts from rock and cave paintings all the way to Hieroglyphics, and all the way to modernism and today’s computer era. As most communication channels are fundamentally insecure; hence, security of visual communication is primarily important, irrespective of the type of visual media. To maintain security, encryption is essential. Historically, encryption has been used to obfuscate secret information and its users were often military leaders or diplomats. The most ancient record of encryption goes back to early Egyptian times, when hieroglyphics were carved in stone. Ancient Babylonians also used intaglio (a collage of images) to differentiate traders. A more developed and systematic cipher was Julius Caesar’s method of substituting alphabets by a certain rule [62]. Since then, many sophisticated methods have been developed. The first image encryption attempt dates back to the 17th century when Dittmers draw an anamorphic picture of king Frederik III and his queen
This picture can be decrypted by the viewing the reflected image in the glass cylinder placed in the centre of the painting.

### 2.5.1 Pseudo-Random Number Generators (PRNGs)

Encryption algorithms frequently employ pseudo-random numbers to achieve desirable properties. For this purpose, PRNGs are employed. A class of PRNGs that recently gained popularity are chaotic PRNGs, which are derived from chaotic dynamic systems. The theory and practice of chaotic cryptography is well defined in [64]. Shannon [39] showed that in the case of secure private key encryption, the length of the message $m$ should not exceed the length of the secret key. In other words, the two parties have to agree on a very long string to be used as the secret key. However, it is not practical to have a secret key of the message length, particularly when the message size increases. Hence, a PRNG is used to generate a keystream the same length as the message size by a shorter string of random bits. Using the PRNG $G$, the two parties need to agree only on a short seed $r$, and exchange the encrypted message $G(r) \oplus m$. Given a probabilistic algorithm, an important question is whether it is provably secure.

The first pseudo-random sequence generator proposed for which one can prove that it is impossible to predict the next number in the sequence from the previous numbers assuming that it is infeasible to invert the RSA function, is due to Shamir [65]. This scheme generates a sequence of numbers rather than a bit sequence. Also, the security proof demonstrates that given the previous numbers output, the adversary is unable to guess the next number. This is not strong enough to prove that, when used in a one-time pad scheme, each bit of the message will be well-protected. Later, Blum and Micali [66] presented the first method for designing provably secure pseudo-random bit sequence generators, based on the use of one-way predicates. Blum and Micali then proposed a particular generator based on the difficulty of computing discrete logarithms. Blum, Blum, and Shub [67] suggested another generator, named the squaring generator, which is simpler to implement and is provably secure assuming that the quadratic residuosity problem is hard. Alexi, Chor, Goldreich, and Schnorr [68] showed that the hardness assumption of the quadratic residuosity problem can be replaced by a weaker assumption that factoring is hard. Next, they suggested a related generator by using the RSA function. Kaliski [69] showed how to extend these methods so that the security of the generator depends on the difficulty of computing elliptic logarithms. Yao [70] showed that a PRNG is perfect if no probabilistic polynomial-time algorithm can distinguish with a probability greater than a non-negligible amount, whether an input string of length $k$ was randomly selected from $\{0,1\}^k$ or it was produced by the PRNG. In other words, if a PRNG passes all polynomial-time statistical tests, then it is perfect [71]. Later, Hastad et al. [72] showed that the existence of a one-way function is equivalent to the existence of a pseudo random bit generator which passes all polynomial time statistical tests.


2.5.2 Encryption Standards

In 1972, the national bureau of standards (NBS), currently known as the NIST, started a program to establish a data encryption standard. In this regard, they announced a call for the design of such an algorithm and in 1974, IBM responded with a design named Lucifer [73]. This design eventually evolved into the DES [50]. DES has a key-length of 56 bits and a block-length of 64 bits. It consists of 16 rounds of what is called a Feistel network. Subsequent to NBS, several other bodies adopted DES as a standard, including the American National Standards Institute (ANSI) and the American Bankers Association. However, DES is not secure, simply because of its relatively short key length, which makes it vulnerable to brute-force attacks. Also, the choice of 64 bits for the block size can lead to security difficulties. Correspondingly, in 1998, the NIST announced a competition for a new block cipher to replace DES with a more secure and a faster cipher which has a block size of 128 rather than 64 bits. Unlike the design of DES, the new algorithm was to be designed in the open and by the public. Fifteen algorithms were submitted to NIST. They came from around the world. A second round narrowed the choice to five of these algorithms and in the summer of 2001, NIST announced the Rijndael algorithm as their choice. The advanced encryption standard (AES) is a variant of Rijndael, where the size of the message block is restricted to 128 bits. This standard was firstly adopted by the United States government and it is currently used worldwide. Unlike its predecessor DES, AES does not use a Feistel network and operates on $4 \times 4$ matrices of bytes, termed the states [51]. The design rationale of AES is based on a substitution-permutation network (SPN), which is fast in both software and hardware.

As a single typical substitution or a single permutation alone does not have much cryptographic strength; so, a substitution-permutation network mixes these properties to form a strong cryptographic algorithm to satisfy Shannon’s confusion and diffusion properties. Such a network adopts the plaintext and the key as inputs, and applies several rounds of substitution and permutation operations to produce the ciphertext. The architecture of these operations is normally based on Boolean operators, such as XOR, modulo addition and bitwise rotation, because they are efficient to perform in hardware. Due to the simplicity and low computation complexity of Boolean operators, they are widely used in designing encryption schemes. In addition, as the structure of an SPN is inherently suitable for parallelism [74], it is hence commonly used as a basic design rationale for many encryption algorithms.

2.5.3 Image Encryption

The recent development in digital multimedia technology has resulted in many useful applications of digital images related to all aspects of life. The majority of such applications communicate over public channels, which are highly insecure. The sensitive visual content can
be easily intercepted while traversing the public channel; hence, the confidentiality of digital images needs to be preserved. Encryption is a solution to this.

Due to the grid structure of digital images, image encryption methods can utilise three types of operations: position permutation, value transformation, and the combination form. The position permutation operations relocate the elements of the original data based on some predefined methods. These methods are simple but they have a low data security. The value transformation operations transform the data value of the original signal with some kinds of transformation. Finally, the combination form performs both position permutation and value transformation. The combination form encryption can offer greater data security.

From the design point of view, the process of image encryption is principally based upon Shannon’s concept of diffusion and confusion [39]. To satisfy this concept, image encryption schemes normally include several rounds of operations to transform the image data into the encrypted domain. Each round includes two major operations: a nonlinear transformation to add confusion and a linear transformation to add diffusion to the input data stream. The non-linear transformations commonly consist of Boolean functions and S-boxes while the linear transformations usually entail permutations. In addition to these operational rounds, image encryption schemes have a key scheduling algorithm which expands a secret key into several round keys.

The encryption operations are normally based on Boolean operators, such as XOR, modular addition and bitwise rotation, because they perform efficiently in hardware. Due to the simplicity and low computation complexity of Boolean operators, they are widely used in designing encryption schemes. The Boolean functions used in image ciphers must satisfy certain properties to ensure resistance against various attacks, such as correlation and algebraic attacks. A Boolean function must have high nonlinearity, high algebraic degree and good correlation immunity to repel correlation attacks [75]. It must also have high algebraic immunity to resist algebraic attacks [75].

A substitution-permutation network consisting of a number of rounds is a common realisation of such principles. The non-linear substitution component, which mixes the secret key bits with those of the plaintext, implies the confusion, and it is primarily used to increase the resistance to linear and differential cryptanalysis. The linear permutation part, which dissipates redundancies, denotes the diffusion, and it is used to increase the computational effort of plaintext attacks. Heys and Tavares [76] have investigated the resistance of these cryptographic networks to differential cryptanalysis and linear cryptanalysis. Additionally, they showed that using large S-boxes with good diffusion characteristics and substituting the permutation between rounds by an appropriate linear transformation is effective in improving the cipher security in relation to these two attacks. Later, Arroyo et al. [77] showed that any one round SPN can be successfully cryptanalysed by chosen-plaintext attacks. They also showed that the security problems of such encryption architecture cannot be resolved just by including a pseudo-random system, such as a chaotic system, as the core of the derived encryption system.
The common weaknesses found in diffusion-confusion algorithms are [78]:

1. The diffusion and confusion architectures are totally disjointed;
2. The control parameters for permutation are fixed in all diffusion-confusion rounds; and
3. In the confusion stage, the key stream extracted from the PRNG only depends on the secret key.

Regarding the first weakness, the attacker can easily divide the diffusion-confusion process into two distinct stages by using a plain-image with identical pixels. Obviously, the permutation process has no effect on these kinds of (monochrome) images and the security of these algorithms only relies on the confusion process. The second weakness is due to the fact that in practical applications, the number of diffusion and confusion rounds is based on the user’s choice. Hence, the total number of control parameters is unknown before the encryption. On the other hand, using different control parameters in each permutation–diffusion round consumes a larger storage. In the confusion process, the keystream is solely determined by the secret key. Hence, decrypting the cipher-image is accomplished by computing the keystream. With regard to this weakness, if the key remains unchanged, then the same keystream is used to encrypt different plain-images. The attacker can therefore obtain the keystream by known-plaintext and chosen-plaintext attacks [79], [80].

Generally, image encryption schemes fall into two main categories: Naïve encryption and selective encryption. Figure 2.8 depicts the logical stages that are taken during these two approaches. The naïve approach operates on the entire bitstream through computationally expensive encryption and decryption stages. However, selective encryption processes a selected part of data through encryption and decryption stages. Table 2.2 summarises the recent work in image encryption.

2.5.3.1 Naïve Approach

The multimedia data can be considered as a binary stream, and therefore, it can be entirely (naïve) encrypted by a standard cryptosystem, such as AES [51] or Salsa20 [81]. In the naïve approach, the data stream can be either bulk encrypted without preserving the syntax format of multimedia data, or it can preserve the format structure, including the header files, and encrypt the content. Bulk encryption by a textual cipher destroys the syntax format of the multimedia data. Hence, encrypted content may not be accessible. Format compliant encryption is suitable for textual or static data, and not for a real-time streaming media. However, it can be employed for protecting the small bitrate multimedia data, which is sent over a high bitrate communication channel. Format compliant encryption enables the post processing of multimedia data in the
Figure 2.8: Stages taken during naïve and selective encryption approaches.
Table 2.2: Summary of the recent work in image encryption

<table>
<thead>
<tr>
<th>Approach</th>
<th>Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bulk encryption</td>
<td>Standard block ciphers</td>
</tr>
<tr>
<td></td>
<td>DES [50]</td>
</tr>
<tr>
<td></td>
<td>AES [51]</td>
</tr>
<tr>
<td></td>
<td>Standard stream ciphers</td>
</tr>
<tr>
<td></td>
<td>Salsa20 [81]</td>
</tr>
<tr>
<td></td>
<td>HC [82]</td>
</tr>
<tr>
<td>Naïve</td>
<td>Fridrich [83]</td>
</tr>
<tr>
<td></td>
<td>Yen and Guo [84]</td>
</tr>
<tr>
<td></td>
<td>Chen et al. [85]</td>
</tr>
<tr>
<td></td>
<td>Guan et al. [86]</td>
</tr>
<tr>
<td></td>
<td>Xiao et al. [87]</td>
</tr>
<tr>
<td></td>
<td>Lian et al. [88]</td>
</tr>
<tr>
<td></td>
<td>Wang et al. [78]</td>
</tr>
<tr>
<td></td>
<td>Zhang and Liu [89]</td>
</tr>
<tr>
<td></td>
<td>Wanq et al. [90]</td>
</tr>
<tr>
<td></td>
<td>Seyedzadeh and Mirzakuchaki [91]</td>
</tr>
<tr>
<td></td>
<td>Fu et al. [92]</td>
</tr>
<tr>
<td></td>
<td>Jolfaei et al. [61], [93], [94]</td>
</tr>
<tr>
<td>Format compliant encryption</td>
<td>Podesser et al. [59]</td>
</tr>
<tr>
<td></td>
<td>Droogenbroeck and Benedett [95]</td>
</tr>
<tr>
<td></td>
<td>Xiang et al. [96]</td>
</tr>
<tr>
<td></td>
<td>Zhu et al. [97]</td>
</tr>
<tr>
<td></td>
<td>Teng and Wang [98]</td>
</tr>
<tr>
<td></td>
<td>Zhang et al. [99]</td>
</tr>
<tr>
<td>Spatial domain</td>
<td>Droogenbroeck and Benedett [95]</td>
</tr>
<tr>
<td></td>
<td>Droogenbroeck [100]</td>
</tr>
<tr>
<td></td>
<td>Stütz et al. [101]</td>
</tr>
<tr>
<td></td>
<td>Xiang et al. [102]</td>
</tr>
<tr>
<td></td>
<td>Yekkala et al. [103]</td>
</tr>
<tr>
<td>Selective</td>
<td>Liu [104]</td>
</tr>
<tr>
<td>Transform domain</td>
<td>Engel and Uhl [105]</td>
</tr>
<tr>
<td>Pre-compression</td>
<td>Sadourny and Conan [106]</td>
</tr>
<tr>
<td>Intra-compression</td>
<td></td>
</tr>
<tr>
<td>Post-compression</td>
<td></td>
</tr>
</tbody>
</table>

encrypted domain, such as editing, scaling, searching and watermark embedding, without decryption. Recently, several format compliant full encryption approaches based on chaotic maps have been proposed for securing the multimedia.
Format compliant full encryption schemes:

- In [83], Fridrich suggested a permutation only block cipher based on a discretised invertible 2D chaotic map. As this cipher was a position permutation-only algorithm, it could not modify the pixel values; hence, Fridrich further extended the discretised map to three dimensions to chaotically substitute the pixel values as well as the position permutation. Under this architecture, permutations and substitutions are performed several times to achieve a satisfactory level of diffusion and confusion. This architecture formed the basic structure of many image encryption techniques presented in the literature.

- In [84], Yen and Guo proposed an encryption method, named chaotic key-based image encryption algorithm (CKBA). This algorithm first generates a pseudo-random binary sequence by a one-dimensional chaotic system. Image pixels are then rearranged according to this binary sequence. Further, rearranged pixels are masked by the selected key. Yen and Guo also suggested a VLSI architecture for this method, which has a low hardware cost and a high computing speed. As such, CKBA is simple but has obvious security flaws. Li and Zheng [107] showed that the complexity of the brute-force attack to this scheme depends on the size of the plain-image, which indicates vulnerability of this scheme for small plain-images. They also showed that CKBA is very weak to plaintext attacks.

- In [85], Chen et al. proposed a two-dimensional (2D) circulation encryption algorithm (TDCEA). This encryption scheme is based on cyclic permutation of each bit-plane using a chaotic sequence generated by a logistic map. To perform the pseudo-random permutation, each set of eight 8-bit elements is considered as an \( 8 \times 8 \) binary matrix being transformed on each row and each column of the matrix by a two bit-circulation function. Chen et al. also suggested a low-cost VLSI architecture for the efficient implementation of TDCEA. However, as TDCEA is a permutation-only cipher, it is insecure against known/chosen-plaintext attacks [108].

- In [86], Guan et al. proposed a chaos-based image encryption scheme using a pixel position permutation by an Arnold cat map and a pixel value substitution by utilising Chen’s chaotic system. However, Xiao et al. [87] analysed Guan et al.’s encryption scheme and showed that this scheme has a number of fundamental flaws. This scheme cannot permute the pixel at position \((0, 0)\). Also, as the permutation and substitution stages of the cipher are independent, a chosen-plaintext attack can be exploited to break the cipher. In addition, the keystream used in the substitution stage is solely dependent on the initial conditions of Chen’s system. This makes the known-plaintext attack natural. To overcome the above-mentioned flaws, Xiao et al. proposed a new cipher by improving Guan et al.’s encryption scheme. In this scheme [87], the sequential process of permutation and substitution is iterated as \( M \) number of rounds \((M > 1)\) to avoid obvious plaintext attacks. Also, after each permutation stage, the position between pixel \((0, 0)\) and \((1, 1)\) is
Chapter 2. Background and State of the Art in 3D Content Encryption Schemes

exchanged. In addition, the keystream of the substitution stage depends on not only the initial conditions of Chen’s system but also the plain-image pixel values.

- In [88], Lian et al. proposed a block cipher based on the composition of three parts: a position permutation by the chaotic standard map, a masking function, and a key generator based on a skew tent map. Compared to multiplication/division and addition/subtraction operations, permutation operations are at a higher cost [88], especially when the plaintext is of large volume. Thus, it is necessary to reduce the computational complexity. To reduce this complexity, Lian et al. suggested computing the permutation mode, which contains the pixels position information, and then permuting the pixels according to the computed mode for \( n \) number of times. To reduce the computational cost of the permutation mode, they also suggested the use of a sine table.

- In [78], Wang et al. proposed an improved diffusion-confusion structure. In this scheme, the control parameters are firstly generated by a relationship between the pixel intensity at the \((0,0)\) position and the output of a logistic map. This means that different plain-images result in distinct control parameters. Therefore, encrypting images with variable control parameters can resist chosen-plaintext attacks because the attacker can only obtain the control parameters of the chosen-image under study. The control parameters are used to permute the pixel positions. Thereafter, the \((0,0)\) position is swapped with position \((r,s)\). Then, another logistic map, which is iterated with a different secret key, produces a pseudo-random key stream to the length of plain-image pixels. Consequently, each cipher-image pixel is produced by masking the pseudo-random key stream with the previously operated cipher-image pixel and the modular sum of the currently operated plain-image pixel with the same pseudo-random key stream. Finally, the first generated cipher-image pixel is exchanged with the last one. The whole process is then iterated a number of times. Increasing the number of rounds can complicate the relationship between plain-image pixels and the secret key. Hence, it can increase the security level, but at the expense of computations and time delays.

- In [89], Zhang and Liu proposed an image encryption scheme by two completely separate and successive parts: a position permutation and a masking procedure. Firstly, pixel positions are shuffled by a permutation box (P-box) generated by a skew tent map. Then, a pseudo-random key stream is generated by a skew tent map. Consequently, each cipher-image pixel is produced by masking the pseudo-random key stream with the previously operated cipher-image pixel and the modular sum of the currently operated plain-image pixel with the same pseudo-random key stream. However, Wang and He [109] showed that this scheme has a number of fundamental flaws, including the low sensitivity to changes of the plain image, fixed permutation vector, weak keys of the skew tent map, and the computer’s finite precision effect. They also showed that Zhang and Liu’s scheme is vulnerable to chosen-plaintext attacks. This weakness is
mainly due to the separability of the permutation and the masking stage. Correspondingly, monochromatic images can reveal the masking key stream.

- In [90], Wang et al. proposed a chaotic image encryption based on the combination of the permutation of pixel positions and changing pixel intensities. In this scheme, the plain-image is shuffled by a discretised Baker map. Then, pixel values are masked by a pseudo-random keystream generated by a combination of the Logistic map, the Sine map, the Cosine map and the Tent map.

- In [91], Seyedzadeh and Mirzakuchaki proposed a chaos-based image encryption scheme based on a diffusion-substitution and a masking process. In this scheme, firstly a pseudo-random key is generated by a coupled 2D piecewise nonlinear chaotic map. Using this key, the pixels array is encrypted in double time, in the forward order and then in the reverse order. Next, pixels are substituted by the s-box of AES. Finally, cipher-image is segmented into four equal size sections and then the masking process is performed on these segments.

- In [92], Fu et al. proposed a new chaotic encryption algorithm based on two separate parts, that is, a bit-level permutation and a masking procedure, to protect real-time transmission of secure medical images over public networks. In the permutation stage, each of the bit-planes is shuffled separately using an Arnold cat map (ACM) with different control parameters. The resultant bit-planes are then combined together to build the permutated image. The permutation process is iterated \( m \) number of rounds. In the next stage, a logistic map is iterated several times to avoid the harmful effect of the transitional procedure. The masking key stream is then generated by the output of the logistic map and a discretising algorithm. The currently operated pixel, the masking key stream, and the previous cipher-pixel are masked. Finally, these successive stages are iterated in several rounds.

- In [93], [94] and [61], Jolfaei et al. suggested using existing and tested ciphers instead of designing new ones for the purpose of syntax-aware image encryption. This approach is more suitable for applications which require a high level of confidentiality rather than efficiency. Jolfaei et al. investigated the application of stream ciphers, including A5/1 [110], W7 [93] and some of the eSTREAM finalists, such as Salsa20 [81] and HC [82], for syntax-aware image encryption. These ciphers work on binary streams. Therefore, to adapt the textual ciphers with the requirement of 2D images, Jolfaei et al. suggested some pre-processing algorithms.
2.5.3.2 Selective Approach

The encryption and decryption procedures are computationally demanding and time consuming, and they are computationally expensive for power constrained real time communication. Thus, encrypting the entire bitstream may not be a suitable solution for high bitrate multimedia, particularly when the transmission is done over a low bitrate communication channel. Basically, the reduction in the computational cost of the encryption process would allow more processes to be executed in parallel.

One solution to this problem is in the use of selective (partial, perceptual or soft) encryption. Selective encryption is a technique which only encrypts a subset of a bitstream. The primary purpose of selective encryption is to reduce the data size (or required memory) and therefore the computational overheads of encryption while attaining an acceptable level of security. A prominent feature of selective encryption is to preserve some functionalities of the original bitstream, for instance, scalability and perceptibility. In this method, the content is separated into two parts: public and private parts. The former is left unchanged and is accessible to all users. The latter is encrypted. Only authorised users have access to the private part. To reduce the memory usage and the computational complexity, the private part is normally kept as small as possible. The exact portion of the public part to the protected part depends upon the target application. In some applications, such as video on demand, Pay-TV and IP network, a lenient visual degradation of encrypted content not only provides users with a low quality (encrypted) version of the multimedia content but also encourages them to pay for the full-quality access to the unencrypted content. In sensitive applications, such as military visual communications, the visual content needs to be completely obfuscated by a strict visual degradation.

Selective encryption can be applied in spatial or transform domain. Each approach has its own limitations and properties. The term spatial domain refers to the multimedia representation itself, and approaches in this category are based on direct manipulation of elements of the multimedia representation. In these methods, the encryption process dissipates the correlation among elements, and thus makes the encrypted data incompressible. Transform domain approaches are denoted by the transformation applied to the multimedia data. For instance, transformation can be interpreted as capturing some form of frequency, and hence the transform domain is referred to as a frequency domain.

Spatial domain techniques increase the computational overheads of the compression process and they reduce the compression ratio due to disturbed statistical properties of the image. Transform domain techniques suffer from mosaic artefacts, which may arise during compression. As such, in an efficient selective encryption scheme, the size of the encrypted code should be similar to that of the original compressed code and the computational overhead added to the compression scheme should be as small as possible.
Due to the problem of transmitting bulky size images with large redundancy over an insecure and bandwidth-constrained communication channel, it is desirable to both compress and encrypt the image data. For the purpose of easy storage and fast transmission, studies on image compression have been performed over a long period of time [111], [112], [113], [114], [115]. Encryption of digital images can be applied before, during or after compression. The implementation of encryption before compression significantly changes the statistical and structural characteristics of the original image, which results in much reduced compressibility. It also introduces much more computational overheads. However, incorporating encryption within or after the compression code induces relatively small computational overheads [104]. In recent years, the need to apply both encryption and compression to digital image communication is rising. A number of image encryption schemes combined with compression have been proposed. Some methods divide the image encryption and image compression into two separate stages [116], [117], [118], [119]. In these schemes, as the encryption and compression stages are completely separated, the adversary does not require consideration of the compression process for the purpose of breaking the cipher. To reduce the computational requirements of data processing and to make the cryptanalysis effort more complicated, some methods were designed by combining the image encryption and the compression in a single process [104], [120], [121], [122], [123].

Selective encryption schemes in spatial domain:

- In [59], Podesser et al. proposed a selective bit plane encryption in spatial domain. They have also investigated the security of their scheme against ciphertext only attacks. The authors showed that encryption of only the Most Significant Bit (MSB) is not secure enough; hence, more bit planes are required to be encrypted. They also showed that encryption of four bit planes (that is 50% of data size) provides high confidentiality.

- In [95], Droogenbroeck and Benedett proposed a selective encryption technique for uncompressed and compressed images. For uncompressed images, they suggested bit level encryption and for compressed ones, they proposed encryption of the sign and magnitude of non-zero DC coefficients and some AC coefficients. Although this technique provides a degraded content, it does not provide a high level security. It is also vulnerable to plaintext attacks [124]. Later, Droogenbroeck [100] extended the idea to a multiple encryption scheme where different sets of DCT coefficients are encrypted by different content owners. This scheme offers several advantages, such as flexibility, multiple encryption, ROI selectivity and format compliance. However, it requires a trade-off between processing power and speed.

- In [96], Xiang et al. proposed a selective image encryption algorithm by masking the image bit-planes with a pseudo-random key stream. The pseudo-random key stream is derived by a one-way coupled map lattice (CML). Based on visual experiments, Xiang et
al. showed that by encrypting at least 4 bit-planes, their scheme can achieve an acceptable security level, which means it can generate uniform cipher-images. Hence, this scheme can selectively encrypt 50% (4 of the 8 bits) of the whole image data to obtain a trade-off between security and performance.

- In [97], Zhu et al. proposed an iterative bit-level permutation and pixel substitution procedure to encrypt images. Because the upper nibbles (4 bits) of pixels carry a higher amount of information, 94.125% of the total information of the image, compared to the lower nibbles, Zhu et al. suggested different strategies for these two groups to reduce the execution time. According to these strategies, the higher 4 bits are permuted individually, while the lower 4 bits are relocated as a whole. Permutation is performed by an Arnold cat map and parameters of this map, which represent the pixel index, are generated by a logistic map. Then, a substitution is performed by masking the pixels with the output of a logistic map.

- In [98], Teng and Wang proposed a spatial image encryption based on a chaotic bit-plane permutation and chaotic masking. This scheme involves three stages. It firstly permutes the higher 4 bit-planes by a CML. Then, it masks the higher 4 bit-planes by the lower 4 bit-planes and a pseudo-random keystream generated by a logistic map. Finally, it permutes the lower 4 bit-planes by the same key stream generated by the logistic map. As stated in [98], bit-level encryption is used to reduce the size of data during encryption. This encryption mechanism is based on a self-adaptive design which uses the lower 4 bit-planes to encrypt the higher 4 bit-planes. Therefore, it is robust to chosen plaintext/ciphertext attacks.

- In [99], Zhang et al. have analysed the inherent characteristics of the bit distributions and the high correlation among bit planes of natural images. They showed that there are strong correlations among the higher bit planes. If bit planes are encrypted independently without such consideration, some security problems may arise. For instance, retrieval of the 7th bit plane can compromise the 8th bit plane, which contains more than 50% of the total information. To overcome this weakness, Zhang et al. proposed an expand-and-shrink strategy to shuffle bit-planes of the image. This strategy is a diffusion-confusion architecture which uses 2D chaotic maps, such as cat map, standard map and baker map, to relocate pixel positions and modify pixel values. In this method, not only the overall bits but also each bit plane is uniformly distributed. Also, the correlations among neighbour bit-planes are dissipated.

Selective encryption schemes in transform domain:

- In [101], Thomas et al. suggested a selective encryption scheme for progressive JPEG modes by encrypting the leading AC coefficients instead of the trailing AC coefficients. Thomas et al. showed that the encryption of the leading coefficients instead of the whole
coefficients has almost similar effect on image quality. Most importantly, it reduces the encryption effort by a large amount.

- In [103], Yekkala et al. proposed a scalable lightweight encryption for images in DCT domain. This algorithm firstly segments the entire image bit stream into a number of blocks and then, encrypts only selected blocks. The selected blocks are the ones that contain edges, and are determined by a threshold value. The rest of the blocks are left unencrypted. The security level can be tuned by changing the threshold value which is used to identify the blocks containing edges.

- In [104], Liu proposed a selective encryption scheme for encrypting JPEG 2000 images. This scheme generates a private initial table using a secret key and a mapping function. Using this table, it encrypts the selected discrete wavelet transform (DWT) code blocks in the entropy coding stage of the JPEG 2000 coding scheme. The advantage of Liu’s encryption scheme is that it does not affect the compressibility and complexity of the standard JPEG 2000 coding scheme.

- In [105], Engel and Uhl proposed a lightweight compression integrated encryption scheme for JPEG 2000 using key-dependent wavelet packets. This approach can be viewed as a header encryption, which encrypts only the information related to the transform domain. Later, Engel et al. [125] analysed this scheme in terms of compression performance, computational complexity and the level of security. They showed that the approach based on the idea of secret wavelet packets cannot be recommended.

- In [102], Xiang et al. proposed a selective scheme, named degradative encryption, before compression by encrypting the unimportant part of image data, such as the least significant bit (LSB) in a bit-plane or high-frequency DCT coefficients. Through this method, details become blurred while the skeleton is kept perceivable. This algorithm is based on the set partitioning in hierarchical trees (SPIHT) [126], and is designed to work in progressive mode for gaining a trade-off between efficiency and security. Using this scheme, users can preview the blurred version of the original image to check whether they are interested in, and only the authorised or paid users can obtain the original image with high quality. Xiang et al. showed that for a secure degradation, it is only required to encrypt less than 10% of data.

- In [106], Sadourny and Conan proposed an encryption scheme after compression for JPSEC (secured JPEG 2000 [127]). JPSEC is a standard framework for the implementation of security tools and services, such as selective encryption, authentication, and integrity. To support selective encryption in JPSEC, Sadourny and Conan suggested a signalling scheme, which embraces two marker components: security components description and code stream security information. The former is used to signal the presence
of protected parts in the bitstream and the latter is employed to signal the information pertaining to the protection method and some integrity data (such as hash values and signatures) of each individual protected part. In this scheme, encryption is applied at the packet level. Each individual packet is encrypted using one or multiple secret keys, relying on symmetric key block cipher algorithms, such as DES or AES. The encrypted images are passed through the entire normal JPEG-2000 production pipeline, resulting in a JPEG-2000 compliant code stream in which some packets are encrypted.

2.5.4 3D Content Encryption

An overview of previous studies in the area of 3D content protection demonstrates that this research is mainly focused on 3D digital watermarking and is in an effort to detect piracy of IPs [128]. However, as a rule of thumb, prevention of unauthorised access to IP precedes its piracy detection. To prevent unauthorised users from accessing valuable 3D content, secure encryption schemes need to be studied. Despite the importance of 3D content encryption, few technological solutions have been given [129], [56], [128], [11], [130], [131].

Based on the communication model of 3D content, encryption is handled by the content provider and decryption is handled by the client. To display the original content, the protected content is decrypted before rendering. In this subsection, we firstly elaborate the basic communication model of 3D content. Then, we overview the prior studies with respect to this model. Based on the literature study, the advantages, disadvantages and limitations of the prior works are identified. Finally, we determine the point cloud representation as the chosen model for the encryption of 3D objects. Reasons for this choice are elaborated.

2.5.4.1 Communication Setup

The basic communication model of 3D content is illustrated in Figure 2.9. This model includes two sides: a content provider and a number of clients (computers) that render and display the 3D content to users. The content provider owns the content and provides the 3D content, including the protected and unprotected content, to the clients. A key management (i.e., key exchange) should happen between the users and the content provider. Key management is not addressed in this work; it is assumed that key exchange takes place over a secure channel. In this model, 3D objects and textures are deposited in the server database on the content provider side. When a user requests for a particular 3D content, the related database is unpacked from the server database and it is stored in the client’s local memory. Afterwards, the 3D data stream is rendered within the graphics pipeline of the client. Finally, the 3D application program displays the rendered 3D content to the user. Users can interact with an object and edit the content, such as texture change and shape deformation. The modifications are sent to the server, which in turn
updates its internal database. Subsequently, the 3D content is reloaded to every client and it will be rendered when users are within the view range of the updated content.

2.5.4.2 Existing 3D Object Encryption Schemes

Table 2.3 compares prior research regarding the 3D content encryption by highlighting the applied methodology, benefits and drawbacks. The comparison indicates that the major problems in 3D content encryption are about point cloud protection, dimensional and spatial stability, and about robustness against surface reconstruction attacks.

In [56], the W3C XML encryption working group (WG) developed an encryption process which considers contents of an XML element as a binary stream and thus, while keeping the syntax format, encrypts 3D content using conventional ciphers. The benefit of this approach is that it is adaptable to different representations, and hence, it does not face the interoperability issues. However, this method is oblivious to the geometry of 3D content and destroys the dimensional and spatial stability of 3D objects.

In [132], Akiyoshi suggested an encryption method for protecting 3D objects in surface-based representations, mainly CAD models. This method firstly alters the geometric shape by distorting one of the coordinate values of the vertices and the equations for the edges or the contours, using affine or nonlinear transformations. Thereafter, it encrypts the 3D data by a non-compliant naive method, such as RSA [135]. This solution has a number of drawbacks. Firstly, the encryption calculations may be costly (in particular when using RSA). In addition, the proposed geometric distortions may not be sufficient to prevent a malicious user from exploiting...
## TABLE 2.3: Comparison among 3D content encryption methods

<table>
<thead>
<tr>
<th>Reference</th>
<th>Approach</th>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML encryption, 2002 [56]</td>
<td>Format compliant naïve encryption</td>
<td>1. Offers granularity; 2. Adaptable to different representations; 3. No interoperability issues.</td>
<td>1. Oblivious to the semantic constraints, such as dimensional and spatial stability, imposed by 3D content; 2. The rendering of encrypted content may corrupt the scene.</td>
</tr>
<tr>
<td>Akiyoshi, 2004 [132]</td>
<td>Encryption of surface-based representations using a geometric shape distortion and a non-compliant naïve encryption</td>
<td>1. Prevents third parties from unauthorised access and modifications to 3D content.</td>
<td>1. Costly calculations; 2. Inefficient geometric distortions; 3. Destroys the syntax; hence, encrypted content may not be readable by content consuming devices.</td>
</tr>
<tr>
<td>Koller et al., 2005 [11]</td>
<td>3D content protection handled by the content provider using a remote rendering server</td>
<td>1. Reduces the startup latency in the client side, and hence displays the protected content quickly.</td>
<td>1. Unable to protect the 3D data stream in applications running in local computers; 2. Vulnerable to surface reconstructions; 3. Unable to protect point cloud vertices.</td>
</tr>
<tr>
<td>Pan et al., 2005 [130]</td>
<td>3D object encryption based on vertex shader programming</td>
<td>1. Complying with the hardware architecture; 2. Security is provided by the infrastructure; hence users do not need to use a separate tool to protect IPs.</td>
<td>1. It is easy to break the cipher and deduce the original 3D data; 2. Inefficient geometric distortions; 3. Encryption of all 3D objects (transparent encryption) may not be necessary.</td>
</tr>
<tr>
<td>Shi et al., 2006 [131]</td>
<td>A digital rights enabled graphics processing system</td>
<td>1. Complying with the hardware architecture; 2. It is difficult to exploit particular properties of a specific 3D object by running software exploits or simple hardware-based tampering; 3. Security is provided by the infrastructure; hence users do not need to use a separate tool to protect IPs.</td>
<td>1. It cannot render the protected scenes with other representation formats due to the interoperability issues.</td>
</tr>
<tr>
<td>Phelps, 2008 [133]</td>
<td>3D object protection by encryption of the 3D data stream and representation of a dummy object in lieu.</td>
<td>1. Informative about the location and boundaries of the encrypted content.</td>
<td>1. Not compatible with the standard file formats; 2. The encrypted data does not respect the syntax of most 3D techniques, and therefore cannot be widely adopted for any rendering devices.</td>
</tr>
<tr>
<td>Éluard et al., 2013 [128]</td>
<td>Vertex coordination shuffling</td>
<td>1. It preserves the dimensional and spatial stability of 3D objects.</td>
<td>1. Vertex coordination shuffling has a low security level.</td>
</tr>
<tr>
<td>Éluard et al., 2014 [134]</td>
<td>Modular addition with the vertex coordinates</td>
<td>1. It preserves the dimensional and spatial stability of 3D objects.</td>
<td>1. It is easy to break the cipher and deduce the original 3D data, because the scheme is vulnerable to plaintext attacks.</td>
</tr>
<tr>
<td>Bitmanagement, 2015 [129]</td>
<td>Non-compliant naïve encryption</td>
<td>1. Prevents third parties from unauthorised access and modifications to 3D content; 2. Utilises standard methods.</td>
<td>1. Does not maintain format compliance; 2. Encrypted content cannot be rendered.</td>
</tr>
</tbody>
</table>
the content. More importantly, non-compliant naïve encryption destroys the syntax; hence, encrypted content may not be readable by a content consuming device, such as a computer or a 3D television.

In [11], Koller et al. investigated several possible protection methods for ensuring the security of the high-resolution geometric details of 3D objects in the underlying 3D graphics system. Based on this investigation, they developed a remote rendering system, which included a 3D viewer client and a rendering server, appropriate for 3D objects’ secure distribution. The rendering server used a number of defensive approaches, such as monitoring and limiting request streams, to protect the 3D geometry from unauthorised extraction. Koller et al.’s approach protects geometric detail of 3D objects by rendering a lower resolution version (fewer polygon surfaces) of 3D models to unauthorised users. In this method, what users see is the snapshots of the rendering results in the rendering server rather than the results rendered with the client’s graphics pipeline. The advantage of this method is that the transmission of sequence of images, rather than 3D models, reduces startup latency on the client side. Thus, protected objects can be displayed quickly. However, as the encryption/decryption process is handled within a remote rendering system, this method cannot be used to protect the 3D data stream in applications running in local computers, such as games. In addition, the method is vulnerable to surface reconstruction attacks [136] because if the server’s 3D graphics system ignores the facets rendering, then the vertices of the 3D object will be revealed. This provides enough information to the potential adversary to reconstruct the surface.

In [130], Pan et al. proposed an encryption method based on vertex shader programming to protect the transmission of 3D models. In this approach, the coordinates of each point are permuted by a simple comparison with the average of the coordinates. Security of such cipher solely relies on a simple permutation of coordinates, and it is therefore easy to break the cipher and deduce the original 3D data. As well as the mentioned design’s shortcoming, this method has a number of drawbacks. Firstly, the proposed distortions may not sufficiently distort the 3D object, and hence the underlying pattern of the encrypted object would still be distinguishable after rendering. In addition, this method encrypts all 3D objects which may not be necessary.

In [131], a digital rights enabled graphics processing system was proposed by Shi et al. In this system, firstly the privacy-sensitive 3D content primitives, including vertices and textures, are encrypted by the content provider. Hence, only valuable graphics data (not all data) are encrypted. The decryption process is then handled within the graphics pipeline, under the control of licenses. The encryption and decryption are performed using AES in cipher block chaining (CBC) or counter mode. This system renders only 3D objects with multi-resolution representations, which have different levels of detail. By this method, protected and unprotected versions of the 3D object primitives can simultaneously be delivered. In this method, it is difficult to exploit particular properties of a specific 3D object by running software exploits or simple hardware-based tampering. Although Shi et al.’s system is a real progress towards secure 3D
environments; it cannot render the protected scenes with other representation formats due to the interoperability issues.

In [133], Phelps described a 3D object protection method by encrypting the privacy sensitive object and representing a dummy object, such as a bounding box, with the same size and the same location in lieu. In this method, the dummy object is stored as non-encrypted data in one file and the protected 3D object as encrypted data in a separate file. Any user may access the non-encrypted data, but only authorised users can access the encrypted data; non-authorised users see the dummy object thereof. This method informs the unauthorised users about the location and boundaries of the encrypted content. Hence, any content modifications that may lead to interference among objects can be avoided. However, this method is not compatible with the standard file formats because it only works with a specific format that stores privacy sensitive objects in two separate files, one file as a non-encrypted data (dummy) and the other one as an encrypted data. Hence, it is only usable by adapted rendering devices, not standard ones. In practical terms, the encrypted data does not respect the syntax of most 3D techniques, and therefore cannot be widely adopted for any rendering devices.

In [128], Éluard et al. proposed a 3D encryption technique based on vertex coordination shuffling to deform and encrypt 3D content. Similarly to Pan et al.’s encryption scheme, Éluard et al.’s proposal is a permutation-only cipher. However, it permutes the set of individual coordinates of vertices rather than permuting the coordinates of each point. A distinct characteristic of the Éluard et al.’s encryption scheme is preserving the geometrical properties of the plaintext object. In other words, it keeps the dimensions of the encrypted object bounded within a box with respect to the bounded dimensions of the plain object. Using this approach, the encrypted object’s dimension cannot exceed the virtual world’s dimensions. Thus, the encrypted object can be rendered without any undesired interference with the objects of the virtual world. However, Éluard et al.’s encryption scheme is not secure because the secret key can be easily deduced by applying all possible permutations to the set of vertices of the encrypted object.

In [134], Éluard et al. proposed a 3D object encryption technique based on a simple modular addition with the vertex coordinates. Similarly to [128], [134] preserves the dimensional and spatial stability of the 3D objects, and thus, it can render the encrypted objects without striking other objects of the virtual world. However, this scheme is not secure because it is vulnerable to plaintext attacks. The pseudo-random sequence used for modular addition can be easily deduced by knowing only one pair of plain and cipher object.

In [129], Bitmanagement developed a server tool for secure dissemination of 3D objects via Internet. In this method, the entire bit stream of 3D graphics data formats, such as the VRML standard, is encrypted by Microsoft’s cryptographic application programming interface (CryptAPI) [137] with 128 bit key length and RC4 [138]. This naïve method is format non-compliant and therefore cannot render encrypted 3D models, such as CAD drawings, without
decryption. However, this method prevents third parties from unauthorised access and modification to 3D content.

2.5.4.3 Point Cloud Encryption: Why?

Over the last decade, point clouds have gained more attention in the graphics community [139], [140] and have been used for several modelling tasks, such as editing [136], and compression [141], and rendering [142]. There are several advantages in comparison with alternatives. Firstly, it is an explicit method of demonstrating the 3D raw data captured by 3D sensors. This representation is also simple and flexible, and does not require any information about connectivity or topological consistency. Secondly, it can be directly used as a rendering primitive, because it circumvents the need for difficult surface construction procedures. Moreover, it is a more data efficient alternative when the available point data density exceeds the viewing screen resolution [143], because there is no need to maintain, store and render the polygons associated with the edge of the off-screen mesh.

A point cloud can also describe the 3D geometry of the surface, including surface position and normal. Hence, the methods applied in point cloud representations can be extended to other surface representations, such as polygonal meshes. In addition, point cloud is a better fit representation for cryptographic applications which require data addition and deformation. It is therefore necessary to encrypt the point cloud before the rendering because if the rendering engine fails to render the surface, then the position of the points and hence the original shape of the object may be revealed. Thus, due to the importance of point cloud protection in many 3D applications, encryption of point cloud representation should be a central part of 3D content security.

2.6 Limitations and Research Challenges

In the context of information creation, processing, transmission and storage, 3D content implies a higher level representation or semantics compared to traditional data and 2D data. Therefore, characteristics of 3D content are significantly different from other types of data. For instance, the data structure, data size, representation, required bitrate, compression complexity, and security requirements may deviate considerably from traditional data and 2D data. These differences introduce several challenges in terms of 3D content encryption. Therefore, the concept of 3D content, including structural elements, representations, standards, formats, communication model and rendering mechanism were reviewed. This review not only gives a clear perspective of the technical requirements of 3D content but also illuminates the limitations imposed by 3D applications. The important requirements of 3D content are format compliance, granularity, semantic constraints, object representation, file format, real-time performance, complexity, compression efficiency, perceptibility and the security level.
3D content encryption is a subclass of data encryption which is principally intended to maintain the functionality of 3D content. It also has a challenging interdisciplinary nature and faces many issues including the requirements of multimedia communication, multimedia retrieval, multimedia compression and hardware resource usage. Due to the nature of 3D content, traditional forms of data security cannot sufficiently address the needs of 3D content security because they mainly process information at the bit-level which is oblivious to the semantics of the information. To design an efficient 3D content cipher, the cryptographic tools including cryptanalysis and statistical methods were thoroughly analysed. Furthermore, the road map to 3D content encryption was drawn.

The study of existing 2D image and 3D object encryption schemes identifies the major challenges in 3D content encryption. It is concluded that the main problems in image encryption are about maintaining the following requirements:

- Format compliance;
- Granularity;
- Real-time performance;
- Computational complexity; and
- Compression efficiency.

In addition, the common weaknesses found in image encryption algorithms are:

- The diffusion and confusion architectures are totally disjointed;
- The control parameters for permutation are fixed in all diffusion-confusion rounds; and
- In the confusion stage, the key stream extracted from the PRNG only depends on the secret key.

The evaluation of previous studies in 3D object encryption demonstrates that the major problems in 3D object encryption are about:

- Point cloud protection;
- Maintaining the dimensional and spatial stability; and
- Robustness against surface reconstruction attacks.

These challenges are addressed in the following chapters.
3

Research Design and Methodology

3.1 Introduction

This chapter gives an overview of the research methodology employed to conduct this study. To attain the research goal, that is 3D content cipher design and evaluation, a quantitative method was undertaken in this research. Although the quest for good ciphers (goal) has always been the holy grail of cryptographers, our aim was to do our best to accomplish this quest for 3D content. The proposed research framework is depicted in Figure 3.1. We started at the current state of knowledge, obtained from a thorough literature review. The critical study of prior works revealed the ‘gap’ in the research undertaken. It also positioned the current study in the context of the previous research. The secondary data obtained from the literature review was then used to delineate the scope of the research. It also helped us to identify the existing problems and limitations. With respect to the problems, we proposed a number of research questions. To address these questions, we constructed testable hypotheses which were derived from secondary data. To verify the correctness of the hypotheses, we developed a research methodology which outlined the appropriate strategy to undertake the research and provided us with the primary data. We used a current theory (secondary data) to test and explore a new theory. The research methodology included two consecutive phases: design, and testing and evaluation phases. The mathematical results (primary data) obtained from the research methods were analysed and then synthesised with the secondary data. Accordingly, conclusions were
Chapter 3. *Research Design and Methodology*

**Research Design**
- Conduct Literature Review
- Identify Research Scope
- Develop Research Methodology
- Design and Development
- Interpret and Report

**Data Analysis and Synthesis**
- Synthesis of Research Findings with Literature
- Draw Conclusions, Limitations and Recommendations

**Publication**
- Write up the Thesis

**Start of PhD**
- Acquire Background Knowledge
- Identify the ‘gap’ in the research
- Identify Research Questions
- Construct Testable Hypotheses
- Develop a Suitable Research Strategy
- Obtain Primary Data

**End of PhD**

*Figure 3.1: Block diagram of the research process.*
drawn. Limitations and recommendations were given. This has modified the current state of knowledge. The final phase of the research process was to write up the thesis.

### 3.2 Research Questions

Based on the literature review and initial investigation presented in the previous chapter, we identified the following research questions:

1. How to design encryption schemes for 3D content, preserving the dimensional and spatial stability.
   
   1.1. How to construct encryption primitives for point clouds using non-isometric deformations.

2. How to prevent disclosure of 3D surface geometry from the texture leakage using a texture encryption scheme that is lightweight and satisfies the security requirement.

3. How to design and evaluate 3D content encryption schemes resistant to well-known attacks.

### 3.3 Hypothesis

The hypothesis attempts to answer the research questions. The answer to the research questions should be sought in the definition of a good 3D content cipher. As a rule of thumb, a good cipher is the one that meets the application requirements. Hence, in the case of 3D content, we looked for the ciphers that meet the 3D requirements (See Subsection 2.4.1). To address the research questions and design a good cipher, the following hypotheses were tested and verified.

1. To address Research Question [1]
   
   - The structure of an iterative product cipher can be used as the basis for developing an efficient encryption scheme for 3D content. This design rationale can meet the diffusion and confusion requirements of Shannon’s concept of a good cipher. The detailed solution is explained in Chapters 4, 5 and 6.
   
   - To meet the dimensional and spatial requirement, random spatial rotations can be utilised to deform the geometry of 3D content. Spatial rotation is a linear transformation represented by an orthogonal matrix. Since the inverse of an orthogonal matrix is its transpose, no extra calculation is required to implement the decryption procedure which is based on the inverse matrix. The detailed solution is explained in Chapters 4, 5 and 6.

   - A secure PRNG can be utilised to uniformly distribute the Euclidean distance among the points of a point cloud. This creates non-isometric deformations by dissipating...
Chapter 3. Research Design and Methodology

the spatial relationships of 3D content. Random distribution of points also makes it difficult to find any pattern inside the unorganised point cloud. This provides resistance to surface reconstruction attacks. The detailed solution is explained in Chapter 4.

- 3D models can be discretised into several geometric primitives, such as polygon soups and meshes. These geometric primitives can be easily converted into point clouds by sampling. Therefore, methods applied on the point cloud representation can be extended to other 3D representations.

2. To address Research Question 2,

- One potential solution to the problem of texture encryption can be in the use of a lightweight encryption scheme with a high level of security, tailored for maintaining the real-time performances. In a true colour (24-bit) representation, 94.125% of the total information is stored in the upper nibbles (4 bit-planes) of the texture image. This suggests employing a strict strategy to encrypt the upper nibbles and a lenient scheme for the encryption of lower nibbles. This approach can improve the encryption performance and can reduce the memory usage. The detailed solution is explained in Chapter 5.

3. To address Research Question 3,

- Security of proposed encryption schemes can be evaluated by the cipher resistance to cryptanalysis, statistical analyses and performance analyses. As well as conventional evaluation methods, proposed ciphers can also be evaluated by methods used in computer graphics and geometry processing, such as the geometry similarity and correspondence between the plain objects and cipher objects. The detailed solution is explained in Chapter 4.

- To measure the robustness of the proposed encryption scheme to surface reconstruction attacks, the distribution of the set of individual coordinates of vertices can be evaluated using the runs test. The detailed solution is explained in Chapter 4.

- Security of the proposed iterated product cipher can be evaluated at each round. It should be analysed what happens if the adversary gains additional access to some of the internal rounds of the computation of the primitive. This gives some insights about the cipher structure and can also identify the optimum number of encryption rounds. The detailed solution is explained in Chapters 4, 5 and 6.

- Cryptanalysis of permutation-only ciphers as well as the cryptanalysis of permutation–substitution ciphers can assist the designer in creating a more robust encryption scheme. The detailed solution is explained in Chapter 6.
3.4 Research Methodology

To maintain the integrity of the research, any ad hoc and unpredictable methods needed to be avoided. Therefore, a reliable and coherent research methodology was conducted. Figure 3.2 demonstrates the diagrammatic outline of the methodology we have undertaken in this research. There are three major phases and the stages in each phase are given in the following subsections. In regards to the hypotheses, cryptosystems were designed. The next step was the testing and evaluation phase. This stage was placed after the design to verify the security and efficiency of proposed ciphers, because cryptosystems must undergo a thorough security evaluation and experimental analysis before practical application. Finally, the results of the evaluation and experimental phase were interpreted and reported.

3.4.1 Initial Investigation Phase

The essence of a system (what it must do) is different from the implementation of the system (how it does what it must do). The application of this notion to the design of 3D content encryption schemes ensures that a problem-centred approach is taken and that the problem is fully understood before any design thinking occurs. The investigation phase involved two components: investigation of technical requirements and investigation of constraints. Technical requirements are factors determined by the problem itself. Constraints are factors that are derived more from the environment of the problem than from the problem itself. For instance, given that the problem was to prevent disclosure of 3D content, transmitting the 3D data in unreadable form was a requirement; maintaining the real-time processing and the renderability of the encrypted data was a constraint.

3.4.1.1 Investigation of Technical Requirements

In this phase of the research, we identified the technical requirements of the problem that we were trying to solve. It should be noted that some requirements were consistent with each other, while other requirements conflicted. Each application has its own specific requirements with different orders of priorities. Therefore, in the design of a 3D content cryptosystem, it is necessary to make a quantitative decision based on conflicting requirements. As a result of this stage, the intended applications and the domain of applicability of 3D content security were determined. The important concepts and requirements of 3D content are format compliance, granularity, semantic constraints, object representation, file format, real-time performance, complexity, compression efficiency, perceptibility and the security level.
Chapter 3. *Research Design and Methodology*

**FIGURE 3.2:** Diagrammatic outline of the methodology.
3.4.1.2 Investigation of Constraints

Constraints are factors that limit the designers’ options but are not mandated by the problem to be solved. Constraints are normally determined by the applicable standards and the implementation platform type. It must be noted that adherence to specific standards can force the use of specific mechanisms. For instance, AES-128 requires the use of a specific 128-bit private-key encryption method. In addition, 3D content cryptosystems must be tailored with respect to the associated constraints on metrics such as the available power, energy, computing ability, area, execution time, and memory requirements, which are all imposed by the simulation and implementation platform.

3.4.2 Design Phase

The investigation phase serves as a statement of the problem to be solved. It also identifies the requirements and constraints limiting the designers’ implementation options. In the design phase, a solution was developed that satisfied the specifications. The following subsections describe the steps within this phase.

3.4.2.1 Investigation of the Encryption Architecture

At this stage, the overall architecture of the encryption scheme was defined. The architecture of modern multimedia ciphers is normally based on the concept of an iterated product cipher which was firstly suggested and analysed by Shannon [39]. A product cipher is constructed by a combination of two or more transformations, including substitution, permutation, and modular arithmetic. The aim of using several transformations is to increase the security in comparison with solely using the individual components, and hence, to increase the robustness to cryptanalysis [38]. To increase the attack complexity and to make the cryptanalysis task cumbersome, product ciphers are normally iterated in multiple rounds, each of which uses a different subkey derived from the original key. Despite the security benefits of increasing the number of operation rounds, it reduces the cipher’s efficiency. Hence, it is important to construct ciphers with an optimum number of operation rounds. Minimal cryptographic structures are built using necessary and sufficient primitives and if eliminated, the encryption scheme becomes insecure. Minimalism is in fact the art of design, where the encryption scheme is stripped down to its most fundamental elements. This issue is an old open problem in cryptography which was initially mentioned by Even and Mansour in [144].

3.4.2.2 Investigation of the Encryption Primitives

This stage defines the suitable encryption primitives required to design and implement the 3D content encryption scheme. There are many primitive operations such as arithmetic operations
and logical bitwise operations, which can constitute the building blocks of the encryption primitives. However, security mainly relies on the mathematical methods these operations are combined together, and hence the choice of operations in them is virtually limitless.

Cryptographic primitives typically stick with operations that are either a single instruction in common architectures or can be implemented cheaply with a combination of a few instructions. Therefore, bit shifts, bit rotations, arithmetic addition, logical bitwise operations like AND, OR, and XOR, are all good candidates for operations. Table lookups are also quite common to speed up implementations. Among the logical bitwise operations, XOR is more common because it is an involution, i.e. a function that is its own inverse. Bitwise-AND and bitwise-OR are not invertible.

From the data structure point of view, 3D content is a composition of a 3D object and a texture image wrapped around it. Each part has a different structure and hence has different requirements, which need to be considered for encryption purposes. To prevent the potential adversary from exploiting the available information, each part needs to be protected. Therefore, 3D content encryption can be considered to be a combination of two separate cryptographic primitives, one for 3D object encryption and the other for texture encryption.

### 3.4.2.2.1 3D Object Encryption

3D objects communicate and interact with their environment by means of a geometric representation with a recognisable boundary. The majority of information is held within 3D geometric shapes. There are truly an endless variety of such shapes, each communicating its own meaning and message. Changing the characteristics of 3D geometries alters the conceptual meaning of 3D objects and conveys different information. Hence, it is important to deform such geometries to protect valuable information from the eye of an unauthorised beholder. However, the geometry deformation should be effective, and it must not violate the semantic constraints of 3D objects. The competitive adversary may attempt to use 3D pattern recognition techniques to find some relationships among the 3D points, and hence, derive hidden patterns. In this research, the question is how to effectively deform such complex geometric objects so that the underlying information will remain unrecognisable. The answer to this problem should be sought in the definition of a good cipher. The encryption process must create a completely different output from the original input. In 3D modelling, this suggests non-isometric deformations. In mathematics, an isometry is a distance-preserving isomorphism between metric spaces. Given a metric space, or a set and scheme for assigning distances between elements of the set, an isometry is a mapping from the elements of the original metric space to the target metric space such that the distance between the elements in the new metric space is equal to the distance between the elements in the original metric space. An isometric deformation of an object in the Euclidean space is thus a deformation which preserves the distance between the vertices. In practice, isometric
deformations only bend the surface without stretching or shrinking it. This means that isometric deformations preserve the intrinsic geometry of the shape which contradicts the definition of a good cipher. Hence, a good 3D content cipher is constructed by non-isometric deformations. The benefit of limiting the design rationale to the class of non-isometric deformations is that it helps to repel any intrinsic geometry similarity between the plain and cipher objects.

In addition to being non-isometric, geometric deformations must not violate the semantic constraints of 3D objects. Without such consideration, encrypted objects may intrude into one another or fall out of the 3D scene. Despite the protection benefits of geometry deformation, it has an adverse impact on graphics processing performance. Encryption process transforms 3D objects into distorted objects, which normally have larger faces than the original objects. Rendering such objects requires more z-buffer updates which is at the cost of computational complexity. This increased complexity can degrade the overall display performance (slower frame rate) for users without credentials. Despite the frame rate penalty for unauthorised users, authorised users can still view the content without drawbacks.

3D objects have many different representation models. Due to the reasons mentioned in Subsection 2.5.4.3 point cloud representation is an appropriate model for applying the encryption process. Hence, the aim of our research is to encrypt point cloud representation.

### 3.4.2.2.2 Texture Map Encryption

Texture images contain intelligible information which is due to the strong correlation among adjacent texels. As each texel is assigned to a particular vertex, texture patterns provide strong cues to the surface orientation, curvature and 3D surface geometry. There is a strong correlation between the geodesic distance between pairs of points on the surface and the distance between corresponding pair of points in the texture image \[145\]. This relationship provides much information about the 3D geometry. Texture leakage may lead to a disclosure of the 3D surface geometry. It is therefore necessary to confuse this relationship by encrypting the texture image. The texture map encryption obfuscates details of the 3D surface but it cannot change the coarse shape of the 3D object.

### 3.4.3 Testing and Evaluation Phase

To test and evaluate a cipher, it needs to be implemented. The implementation translates the design into reality. The proposed cipher will be implemented using a Matlab code on a machine with Intel Core i7 2.5 GHz processor and 16 GB of installed memory running under Windows 7. The testing and evaluation phase evaluates the security of the cipher design and tests its statistical properties.

Security is the foremost concern in a cryptosystem and it is normally difficult to assess. Security of a cipher is supported by convincing evidence obtained from cryptanalytic tools and
statistical analyses. In other words, a cipher should be thoroughly analysed before being used in practice. In addition to security evaluation, the encryption performance is also an important factor to consider, especially for real-time applications which require high level of efficiency. Conclusions found in the literature are important clues to develop testing and analysis methods and then to evaluate these methods.

### 3.4.3.1 Cryptanalysis

Cryptanalytic tools evaluate the cipher based on different scenarios to check whether the computational cost of deducing the secret key can be reduced from that of exhaustive search. Cryptanalytic tools can also assist in identifying any shortcomings in the cipher design. These tools are listed as follows:

- Ciphertext-only attack;
- Known-plaintext attack;
- Chosen-plaintext attack;
- Chosen-ciphertext attack; and
- Chosen-key attack.

From the cryptographic point of view, a strong encryption scheme should resist all of the above-mentioned attacks. The detailed description of the cryptanalytic attacks is given in Subsection 2.3.2.2.

### 3.4.3.2 Statistical Analysis

Statistical analyses evaluate the statistical properties of an output stream of a cipher, independent of knowledge of the cipher structure. The failure in statistical tests indicates a bias in the cipher output, and hence, shows that it can be predicted from input. However, having good statistical properties alone cannot guarantee the cryptographic security. Statistical analyses are complementary to cryptanalysis. In the literature, there a number of commonly used statistical metrics, such as:

- Key space analysis;
- Key sensitivity analysis;
- Plaintext sensitivity; and
- Randomness test.

A detailed description of the above-mentioned analyses is given in Subsection 2.3.2.3.
3.4.3.3 Performance Analysis

Performance analysis is very important for practical use of any cipher. Performance of encryption normally depends on the simulation (implementation) environment including software and hardware platforms, and the optimisation level of the simulation code of the design. The modules of the software platform which have performance impact on encryption are the programming language and the operating system, and those of the hardware platform are the structure of central processing unit, memory size, number of instructions, number of clocks per instruction, number of equivalent gates, latency, and power consumption per bit. Performance of a cipher is normally measured by calculating the encryption time, encryption speed in clocks per byte, computational complexity, and space complexity.

The encryption time is considered as the processing time that an encryption algorithm takes to convert a plaintext into a ciphertext. To have an accurate benchmark result, each timing test should be executed several times on the same benchmarking system. As an absolute requirement for designing a good cipher, the processing time to encrypt \( n \) bytes should be asymptotically proportional to \( n \). Encryption time is normally used to calculate the throughput of an encryption scheme. It indicates the speed of encryption. The throughput of the encryption scheme is calculated as the total plaintext in bytes encrypted divided by the encryption time. Measuring bytes per second is a useful measure but it is heavily dependent on the benchmarking system. In other words, it can only show the performance of an encryption algorithm on a single machine and gives no real indication of its performance on other machines. Therefore, to allow applicable comparisons, encryption speed is calculated in clocks per byte which is the number of required processor clock cycles to process each byte. The number of CPU clock cycles is a metric reflecting the amount of energy consumption of the CPU while operating on encryption operations.

The computational complexity estimates the number of elementary operations executed by an algorithm, where each elementary operation consumes a fixed amount of time to accomplish a task. The computational complexity of an algorithm is usually articulated by big \( O \) notation. This notation excludes coefficients and lower order terms (asymptotically upper bound). The computational complexity is expressed asymptotically, as the input size approaches infinity. As the time performance of an algorithm may vary with different inputs of the same size, the computational complexity of an algorithm is normally considered by the worst case, denoted by \( T(n) \), which is defined as the maximum amount of time consumed by any input of size \( n \). Computational complexities are classified by the nature of the function \( T(n) \). For example, an algorithm with \( T(n) = O(2n) \) is known as an exponential time algorithm, an algorithm with \( T(n) = O(\log n) \) is named a logarithmic time algorithm, and an algorithm with \( T(n) = O(n) \) is denoted as a linear time algorithm.
Apart from the computational complexity, space complexity is also an important measure which estimates the number of memory cells that an algorithm requires to perform a specific task. A good algorithm keeps this number as small as possible. There is often a time-space trade-off involved in a problem, because it may not be possible to solve the algorithm with small computation time and low memory consumption. Therefore, the algorithm designer has to make a compromise and exchange the computation time for memory consumption or vice versa.

3.5 Conclusion

This chapter presented our research questions together with the relevant hypotheses. The research methodology justifying the hypotheses was discussed. We have further given a description of and a motivation for our own design approach. For each of the different steps of the design and analysis, we have highlighted our approach and indicated the origin of the initial concepts. The detailed explanation of our proposed methods are adjourned to Chapters 4, 5 and 6.
3D Object Encryption Which Maintains Dimensional and Spatial Stability

4.1 Introduction

Advances of multimedia computing and networking have unlocked the path for the application of 3D objects in a variety of domains, including virtual reality and augmented reality. The fast growing demand for high definition visualisation applications has opened up a number of challenges regarding the confidentiality of 3D objects. Secure communication of 3D objects is a legitimate concern of IP owners, developers, government regulatory bodies and law enforcement agencies. Thus, there is a strong need to protect 3D objects against unauthorised use or other security violations. To maintain confidentiality of 3D objects, encryption is essential. In this chapter, we propose a technical solution for the problem of 3D object encryption which overcomes the limitations of the current techniques in addressing the confidentiality requirement of 3D objects.

It seems natural to use established and tested ciphers to encrypt 3D objects bit by bit. However, due to special features of 3D objects, naïve encryption may not be a suitable solution for many 3D applications. The problem of 3D object encryption is beyond the application of established and well-known encryption algorithms. This is primarily due to the structure of 3D objects and the way they are used commercially. Unlike data encryption, where a complete bitstream is encrypted, 3D content encryption introduces several challenges.
In comparison with 1D and 2D data, 3D objects imply a higher level representation or semantics. 3D objects have a 3D geometry and in many applications, such as socialising metaverses and games, there is a requirement to display such geometry in a 3D space. Hence, 3D objects should be confined to a virtual space in order to be displayed. If such consideration is not taken into account, then encrypted objects may exceed the viewing screen resolution or they may overlap with other objects of the virtual world. Hence, the encryption outcome may conceal other objects behind its surface. This decreases users’ observation capability. Conventional ciphers, such as AES, are oblivious to dimensional and spatial stability of 3D objects and hence, the rendering may spill out the encrypted outcome from the intended size and location, and corrupt the whole 3D scene. In practical terms, conventional ciphers destroy the spatial and dimensional stability of 3D objects. It can be argued that the encrypted content may not require rendering and it would be better to display nothing. This may not be an ideal solution. In many applications, such as in virtual museums and 3D e-commerce, if users have no means of noticing that something is missing, it is then unlikely that they will find that something is missing. In practical terms, the motivation to pay for having access to 3D content would be lacking.

An example of the dimensional and spatial stability requirement would be the case of a game production workflow where a crew of level designers, artists and programmers cooperate under the supervision of a producer. The output of the production workflow is a game program and a series of game assets to be consumed by the game program. These assets include 3D models and images data. The 3D designs are normally updated during the production workflow. For instance, some parts or textures can be added to or added from a particular object. In the production workflow, the design studio may not be willing to give access to the full scene to a particular crew. However, this crew may need to have access to visual cues in order to insert graphical elements without striking existing ones that would be invisible because of the protection framework.

3D objects have different kinds of representations depending on the type of 3D scanning device used. For instance, 3D laser scanners produce a cloud of points while CT scanners create a volumetric model in the form of a 3D cube. In addition, different 3D applications, such as medical applications and CAD, use different representations. In this chapter, we focus on point cloud model. Over the last decade, point clouds have gained more attention and have been used for a multitude of modeling tasks, such as editing and compression. The reason for this uptake of the point cloud representation is that it offers several advantages compared to other 3D object representations. Firstly, it is an explicit method of demonstrating the 3D raw data captured by 3D sensors. This representation is also simple and flexible, and does not require any information about connectivity or topological consistency. Hence, it can be directly used as a rendering primitive to circumvent the need for difficult surface construction procedures. Moreover, point cloud representation is very efficient when the available point data density exceeds the viewing screen resolution, because there is no need to maintain, store
and render the polygons associated with the edge of the off-screen mesh. More importantly, other 3D representations, such as polygonal meshes, can be easily converted into point clouds by sampling. Therefore, methods applied to the point cloud representation can be extended to other 3D representations.

In this chapter, we propose a chaos-based symmetric encryption scheme for protecting 3D objects for two reasons. Firstly, due to the large amount of data involved in a 3D object, it requires a huge storage capacity and transmission bandwidth. To provide a better execution performance, we consider symmetric (rather than asymmetric) encryption. Secondly, chaos-based cryptographic primitives have a number of advantages as shown in [64], such as the sensitivity to input data. Chaos has been used to design encryption schemes for non-3D multimedia data [148], such as image and video. However, these image and video encryption schemes are not applicable to 3D object encryption due to the 3D geometry and representation, as explained in the above discussion.

The proposed cipher employs random permutation matrices and random geometric rotations to deform the geometry of 3D objects. Permutation and rotation are linear transformations represented by orthogonal matrices. Since the inverse of an orthogonal matrix is its transpose, no extra calculation is required to implement the decryption procedure which is based on the inverse matrix. This remarkable property makes the implementation of decryption very efficient.

An overview of previous studies in the area of 3D object protection (as detailed in Chapter 2) demonstrates that this research is mainly focused on 3D digital watermarking and is in an effort to pinpoint the source of leaks (traitor tracing) [149], [150]. However, digital watermarking is a complement to encryption and one can never be used to replace the other. To prevent unauthorised users from accessing valuable 3D content, secure encryption schemes need to be studied. Despite the importance of 3D content encryption, few technological solutions have been given [11], [130], [131], [133], [128]. However, the method proposed in [11] is not applicable to point cloud representation and may leak the point cloud information, and the methods proposed in [130] and [128] are not secure as they leak the point cloud information and cannot resist surface reconstruction attacks. In addition, the method suggested in [131] cannot maintain dimensional and spatial stability, and the method proposed in [133] is not compatible with standard file formats.

Following the above discussion, this study addresses the major shortcomings of the literature, and gives a technical solution to the problem of 3D content encryption, which encrypts 3D point cloud based objects making use of pseudorandom permutations and geometric rotations. The proposed cipher is compatible with standard file formats and maintains the semantic requirements of 3D objects, including the dimensional and spatial stability. We showed that the proposed cipher has a large key space and so is robust against brute-force attack. The rigorous security analysis showed no statistical weaknesses in the cipher and demonstrated no simple method of recovering the secret key. It also confirmed the security of the proposed cipher against
known/chosen plaintext attacks and surface reconstruction attacks. The sensitivity analyses indicated that the proposed cipher is highly sensitive to the changes of the plaintext and secret key. Moreover, a spatial randomness test determined that there is no presence of homogenous patterns in the cipher point clouds. In addition to security evaluations, a performance analysis was performed to evaluate the encryption speed of the proposed cipher.

The remainder of this chapter is organised as follows: Section 4.2 provides details of the proposed encryption and decryption schemes. Section 4.3 proves that the proposed scheme ensures the dimensional and spatial stability of the original content. Sections 4.4 and 4.5 evaluate the security of the proposed cipher using cryptanalysis and statistical methods, respectively. Section 4.6 measures the performance of the proposed cipher by calculating its computational complexity and encryption/decryption time. Finally, Section 4.7 concludes that the proposed scheme is secure, efficient and feasible.

### 4.2 Proposed 3D Object Encryption Scheme

In this section, the encryption and decryption procedures of the proposed cipher will be described. Before the algorithm description, we firstly explain the content-dependent metadata employed to ensure the semantic requirements of 3D point clouds. 3D point clouds must be placed in designated locations and must be confined in limited spaces with defined boundaries. Hence, we assume that every point cloud is distributed in a bounding sphere of radius $r$ with center $G$, where, $G$ is the barycentre of the point cloud and $r$ is the distance from the farthest vertex to $G$. In certain 3D content applications, such as animation and game production workflow, the unauthorised designers may need to know the size and position of objects in order to insert graphical elements without striking existing ones that would be invisible because of the protection framework. Hence, $r$ and $G$ need to be discernible to both authorised and unauthorised users. We therefore keep $r$ and $G$ as public information and obfuscate privacy-sensitive information, that is, point cloud vertices, with respect to these public content-dependent metadata.

To elaborate the steps of the encryption algorithm, let $P_1, P_2, \ldots, P_n$ be points of the plain point cloud, and $C_1, C_2, \ldots, C_n$ be points of the corresponding cipher point cloud. For any $j$ $(1 \leq j \leq n)$, $P_j$ and $C_j$ are defined as follows:

\[
P^j = \begin{bmatrix} p^j_1 \\ p^j_2 \\ p^j_3 \end{bmatrix} \in \mathbb{R}^3, \tag{4.1}
\]

\[
C^j = \begin{bmatrix} c^j_1 \\ c^j_2 \\ c^j_3 \end{bmatrix} \in \mathbb{R}^3. \tag{4.2}
\]
Chapter 4. 3D Object Encryption Which Maintains Dimensional and Spatial Stability

To achieve the confusion and diffusion properties defined by Shannon [39], and to maintain the dimensional and spatial stability, we propose an \( n \)-round encryption scheme based on a combination of a confusion-diffusion structure and 3D geometric rotations. As we will analyse later in Section 4.4, two rounds of encryption is enough to have a secure cipher. More rounds of encryption give a higher level of security at the price of losing efficiency. Without loss of generality, in the following subsections, the components used in the encryption algorithm as well as the detailed description of the two-round encryption algorithm will be given. To illustrate the encryption steps, a block diagram of the proposed encryption algorithm is depicted in Figure 4.1.

![Figure 4.1: Structure of the encryption algorithm](image)

4.2.1 Key Scheduling Algorithm

To encrypt a large number of points, we need to iterate the encryption operations several times. Therefore, we first present a key scheduling algorithm based on a Chebyshev map [151] to expand the relatively short secret key to a large expanded key. To avoid simple relationships between the secret key and the expanded key stream and to resist certain types of cryptanalysis, such as related-key attacks and slide attacks, the key schedule algorithm produces different key streams for different encryption rounds [152].

The key scheduling algorithm is described as follows:

\[
k_i^j = \cos(D\cos^{-1}(k_i^{j-1})), \quad \text{for } 1 \leq i \leq 6 \text{ and } 1 \leq j \leq 2n,
\]

(4.3)

where \( D \) is a constant denoting the degree of the Chebyshev map, \( K = (k_1^0, k_2^0, k_3^0, k_4^0, k_5^0, k_6^0) \) denotes the encryption seed point (secret key), and for any \( i \) (1 \( \leq i \leq 6 \)), \( k_i^0 \in [-1, +1] \).
4.2.2 Pseudorandom Point Generation Process

To meet the confusion requirements, the relationship between the encryption key and the cipher point cloud should be made as complex as possible. To this end, in each encryption round, \( n \) pseudorandom points are generated within the point cloud’s sphere of radius \( \sqrt{3}r \) with center \( G \), as follows:

\[
O^j_v = G + r \cdot A^j_v, \quad \text{for } 1 \leq v \leq 2 \text{ and } 1 \leq j \leq n,
\]

(4.4)

where \( v \) represents the encryption round, \( j \) represents the point index, and \( A^t \) denotes the transpose of matrix \( A \). \((k^0_{1}, k^0_{2}, k^0_{3})\) is the seed point for this point generation process.

4.2.3 Permutation Process

To meet the diffusion requirements, the statistical relationship between the plain and cipher point clouds should be made as complex as possible. To this end, the dimensional coordinates of points of the plain point cloud are shuffled with the coordinates of pseudorandom 3D points. This dissipates any meaningful relationship between the points of the plain point cloud. Permutation of a large number of coordinates, for instance, all of the dimensional coordinates at once, may not be an efficient approach because producing a large-scale permutation matrix requires a considerable amount of computation, time and memory. In order to increase the permutation efficiency of the encryption scheme, permutations are locally employed to reorder subsets of the set of all dimensional coordinates of the plain-points and pseudorandom points specified in the previous subsection. The following definitions are given for any given round of the encryption scheme to elaborate the permutation process.

**Definition 4.1.** Let \( \pi = \{k^1_1, \ldots, k^n_1, k^1_2, \ldots, k^n_2, k^1_3, \ldots, k^n_3, k^1_4, \ldots, k^n_4, k^1_5, \ldots, k^n_5, k^1_6, \ldots, k^n_6\} \) denote the permutation keystream.

**Definition 4.2.** Given \( n \) plain-points \( P^j \) and \( n \) pseudorandom points \( O^j \), \( j = 1, \ldots, n \), the universal set \( U \) is defined as the set of all dimensional coordinates of \( P^j \) and \( O^j \), that is,

\[
U = \left\{ u_k \mid u_{i+3(j-1)} = p^j_i \text{ and } u_{i+3(j-1)+3n} = o^j_i, \quad \text{for } 1 \leq i \leq n \text{ and } 1 \leq j \leq n \right\}. \quad (4.5)
\]

By definition above, the cardinality of the universal set \( \#U = 6n \).

**Remark 4.1.** Given an input array of size \( 6n \), there are \( (6n)! \) possible permutations for the inputs. To sort this input, any deterministic comparison-based sorting algorithm requires performing \( O(n \cdot \log n) \) comparisons in the worst case [153].

To reduce this complexity to \( O(n) \) and perform an efficient permutation, the universal set is first partitioned into \( \left\lfloor \frac{6n}{5} \right\rfloor \) small subsets defined as follows:
Definition 4.3. For any \( 1 < m < \left\lfloor \frac{n}{8} \right\rfloor \), let \( X_m \) denote a subset of the universal set \( U \), which is defined as follows:

\[
X_m = \left\{ x_k \mid x_{i+3(j-1)} = p_i^{j+8(m-1)} \text{ and } x_{i+3(j-1)+24} = o_i^{j+8(m-1)}, \text{ for } 1 \leq i \leq 3 \text{ and } 1 \leq j \leq 8 \right\}. \tag{4.6}
\]

If \( 8 \nmid n \), the last \( l \) plain-points and \( l \) pseudorandom points remain unclassed, where \( l = n - 8\left\lfloor \frac{n}{8} \right\rfloor \).

In this case, these elements are added to the last subset \( X_{\left\lfloor \frac{n}{8} \right\rfloor} \), which is constructed as follows:

\[
X_{\left\lfloor \frac{n}{8} \right\rfloor} = \left\{ x_k \mid x_{i+3(j-1)} = p_i^{j+(n-1)-8} \text{ and } x_{i+3(j-1)+24} = o_i^{j+(n-1)-8}, \text{ for } 1 \leq i \leq 3 \text{ and } 1 \leq j \leq 8+l \right\}, \tag{4.7}
\]

where \( l = n - 8\left\lfloor \frac{n}{8} \right\rfloor \). For any \( m (1 \leq m < \left\lfloor \frac{n}{8} \right\rfloor) \), by definition the cardinality of each subset \( \#(X_m) = 48 \). If \( 8 \mid n \), then the cardinality of the last subset \( \# \left( X_{\left\lfloor \frac{n}{8} \right\rfloor} \right) = 48 \); otherwise \( \# \left( X_{\left\lfloor \frac{n}{8} \right\rfloor} \right) = 48 + 6l \). Obviously, by definition the following conditions hold for the subsets \( X_m \):

\[
\bigcup_{m=1}^{\left\lfloor \frac{n}{8} \right\rfloor} X_m = U, \tag{4.8}
\]

and

\[
X_m \bigcap X_n = \emptyset , \text{ for } m \neq n. \tag{4.9}
\]

Remark 4.2. For any \( m (1 \leq m < \left\lfloor \frac{n}{8} \right\rfloor) \), permutations are performed locally in each subset \( X_m \), where \( 48 \leq \#(X_m) \leq 90 \). This makes the permutation more efficient because for each subset \( X_m \), there are \( \#(X_m)! \) possible permutations, and sorting the elements of each subset \( X_m \) requires \( O(1) \) comparisons. Therefore, the computational complexity of rearranging the elements in all of the \( \left\lfloor \frac{n}{8} \right\rfloor \) subsets is \( O(1) \). This is more efficient than the permutation of the whole \( 6n \) elements at once.

Remark 4.3. The design rationale for specifying a particular size for the subsets is to achieve resistance to known/chosen plaintext attacks by two rounds of encryption. To clarify further, see Section 4.4 for the detail of cryptanalysis.

Utilising the definitions above, the permutation process is defined as follows. For any \( m (1 \leq m < \left\lfloor \frac{n}{8} \right\rfloor) \), \( \Pi_{\left\lfloor \frac{n}{8} \right\rfloor}^{48\left(m-1\right)+48} : X_m \rightarrow X'_m \), where

\[
X'_m = \left\{ x'_j \mid x'_j \in X_m, x'_j = \Pi_{\left\lfloor \frac{n}{8} \right\rfloor}(x_j), \text{ for } 1 \leq j \leq 48 \right\}. \tag{4.10}
\]
If $8 \mid n$, then for $m = \lfloor \frac{n}{8} \rfloor$ the same mapping is applied; Otherwise, for $l \ (1 \leq l \leq 7)$, $8 \mid n - l$, \[ \Pi_{\lfloor \frac{n}{6(n-1)} \rfloor - 1}^{\lfloor \frac{n}{6(n-1)} \rfloor} : X_{\lfloor \frac{n}{8} \rfloor} \rightarrow X_{\lfloor \frac{n}{8} \rfloor}, \]

\[ X'_{\lfloor \frac{n}{8} \rfloor} = \left\{ x'_j \mid x'_j \in X_{\lfloor \frac{n}{8} \rfloor}, x'_j = \Pi_{\lfloor \frac{n}{8} \rfloor}(x_j), \text{ for } 1 \leq j \leq 48 + l \right\}. \quad (4.11) \]

The permutation outcome, including the permuted plain point cloud $P' = \{P^{t1}, P^{t2}, \ldots, P^{tn}\}$ and the permuted pseudorandom points $O' = \{O^{t1}, O^{t2}, \ldots, O^{tn}\}$, are obtained as follows. For any $m \ (1 \leq m < \lfloor \frac{n}{8} \rfloor)$,

\[ P'^{tj+8(m-1)} = \left( x'_{3(j-1)+1}, x'_{3(j-1)+2}, x'_{3(j-1)+3} \right)^t, \quad \text{for } 1 \leq j \leq 8, \quad (4.12) \]

\[ O'^{(j-8)+8(m-1)} = \left( x'_{3(j-1)+1}, x'_{3(j-1)+2}, x'_{3(j-1)+3} \right)^t, \quad \text{for } 9 \leq j \leq 16. \quad (4.13) \]

If $8 \mid n$, then the above relations hold for $m = \lfloor \frac{n}{8} \rfloor$. If for $l \ (1 \leq l \leq 7)$, $8 \mid n - l$, then for $m = \lfloor \frac{n}{8} \rfloor$, the permuted plain point cloud and the permuted pseudorandom points are obtained as follows:

\[ P'^{tj+(n-l)-8} = \left( x'_{3(j-1)+1}, x'_{3(j-1)+2}, x'_{3(j-1)+3} \right)^t, \quad \text{for } 1 \leq j \leq 8 + l, \quad (4.14) \]

\[ O'^{(j-8)+(n-l)-8} = \left( x'_{3(j-1)+1}, x'_{3(j-1)+2}, x'_{3(j-1)+3} \right)^t, \quad \text{for } 9 + l \leq j \leq 16 + 2l. \quad (4.15) \]

### 4.2.4 Geometric Rotation Process

To maintain the dimensional and spatial stability, and also to safeguard the confusion-diffusion process from chosen-plaintext attacks, such as point re-ordering attacks [149], for any $j \ (1 \leq j \leq n)$, the $j$-th point of $P'$ is geometrically rotated about the $j$-th point of $O'$ with random Euler angles $\left( \alpha'_1, \alpha'_2, \alpha'_3 \right)$. As the inverse of a 3D rotation matrix is equal to its transpose, no extra calculation is required to compute the reciprocal matrix. This remarkable property in the design of our cryptographic algorithm makes the implementation of decryption very efficient. The geometric rotation function $\mathcal{R}ot(\cdot)$ is defined as follows:

\[ \mathcal{R}ot \left( P'^{tj} \right) = \psi \cdot R^j \left( \alpha'_1, \alpha'_2, \alpha'_3 \right) \times \left[ P'^{tj} - O'^{tj} \right] + O'^{tj}, \quad \text{for } 1 \leq j \leq n, \quad (4.16) \]

where $K = \left( k^0_1, k^0_2, k^0_3 \right)$ denotes the seed point (secret key), and $R \left( \alpha_1, \alpha_2, \alpha_3 \right)$ is the 3D rotation matrix defined as follows:

\[
\begin{bmatrix}
\cos(\alpha_2) \cos(\alpha_3) & \sin(\alpha_1) \sin(\alpha_2) \cos(\alpha_3) - \cos(\alpha_1) \sin(\alpha_3) & \cos(\alpha_1) \sin(\alpha_2) \cos(\alpha_3) + \sin(\alpha_1) \sin(\alpha_3) \\
\cos(\alpha_2) \sin(\alpha_3) & \sin(\alpha_1) \sin(\alpha_2) \sin(\alpha_3) + \cos(\alpha_1) \cos(\alpha_3) & \cos(\alpha_1) \sin(\alpha_2) \sin(\alpha_3) - \sin(\alpha_1) \cos(\alpha_3) \\
-\sin(\alpha_2) & \sin(\alpha_1) \cos(\alpha_2) & \cos(\alpha_1) \cos(\alpha_2)
\end{bmatrix}. \quad (4.17)
\]
Algorithm 4.1 Pseudo-code of the encryption algorithm

1: procedure ENCRYPTION($P, r, G, ψ, K$) 
2: \{Encryption computes the cipher-points $C^1, C^2, \ldots, C^n$, given the plain-points $P^1, P^2, \ldots, P^n$, content-dependent metadata including $r$ and $G$, control parameter $ψ$, and the secret key $K$\}
3: Generate an expanded key stream using the key scheduling algorithm, as explained in Subsection 4.2.1.
4: for Round $← 1, 2$ do
5:   Generate $O^1, O^2, \ldots, O^n$ using the pseudorandom point generation algorithm explained in Subsection 4.2.2.
6:   for $m ← 1, \lfloor \frac{n}{8} \rfloor$ do
7:     $Π : X'_m ← X_m$, as explained in Subsection 4.2.3
8:   for $j ← 1, 8$ do
9:     Compute $C^j$ by geometric rotation of $P'^j$ about $O'^j$, as explained in Subsection 4.2.4
10:   $P^j ← C^j$.
11: end for
12: end for
13: end procedure

$0 < ψ \leq \frac{1}{9}$ is the scaling factor which is used to adjust the size of the rotated point cloud. For any encryption round $l$ ($1 \leq l \leq 2$), Euler angles are obtained as follows:

$$α_{i,j}^{i+\lfloor \frac{j}{2} \rfloor,n} = \lfloor 180k_{i+\frac{j}{2}} \rfloor, \text{ for } 1 \leq i \leq 3 \text{ and } 1 \leq j \leq n. \quad (4.18)$$

4.2.5 Encryption and Decryption Algorithms

Details of the proposed encryption and decryption algorithms are described as pseudo-codes in Algorithm 1 and Algorithm 2, respectively. The proposed cipher is based on a series of permutations and rotations. Both permutation and rotation are linear transformations represented by orthogonal matrices. Since the inverse of an orthogonal matrix is its transpose, no extra calculation is required to implement the decryption procedure which is based on the inverse matrix.

4.3 Dimensional and Spatial Stability

A principal requirement for 3D content encryption is to ensure the dimensional and spatial stability of the original content. If such consideration is not taken into account, then encrypted objects may exceed the viewing screen resolution or they may collide with other objects of the virtual world. The literature review indicates that previous dimension and space preserving encryption schemes, such as [133] and [128], attempted to provide stability in the form of a bounding box. However, from the cryptographic point of view, providing stability in the form of a bounding box is not an appropriate method for the encryption applications as it discloses...
Algorithm 4.2 Pseudo-code of the decryption algorithm

1: procedure \textsc{Decryption}(C, r, G, ψ, K)
\{
\begin{itemize}
  \item Decryption computes the plain-points \( P^1, P^2, \ldots, P^n \), given the cipher-points \( C^1, C^2, \ldots, C^n \), content-dependent metadata including \( r \) and \( G \), control parameter \( ψ \), and the secret key \( K \).
\end{itemize}
\}

2: Generate an expanded key stream using the key scheduling algorithm, as explained in Subsection 4.2.1.

3: \textbf{for} \( \text{Round} \leftarrow 2, 1 \) \textbf{do}

4: \textbf{for} \( m \leftarrow 1, \left\lceil \frac{n}{8} \right\rceil \) \textbf{do}

5: \( \Pi : \{ \{ P^{j+8(m-1)} \}_{j=1}^8 \}, \{ O^{j+8(m-1)} \}_{j=1}^8 \} \leftarrow \{ \{ P^{j+8(m-1)} \}_{j=1}^8 \}, \{ O^{j+8(m-1)} \}_{j=1}^8 \}, \) as explained in Subsection 4.2.3.

6: \textbf{end for}

7: \textbf{end for}

8: \textbf{end procedure}

The maximum dimensional coordinates of the plain point cloud. Therefore, we maintain the stability via bounding spheres, in which the maximum dimensional coordinates of the plain point clouds are not revealed. Given the radius of the bounding sphere, the adversary cannot correctly decompose it into vertex coordinates. Also, the size of the bounding sphere of the encrypted point cloud is adjustable, in that it can be made strictly smaller, but not greater than the size of the bounding sphere of the plain point cloud. This can improve the usability of the encrypted point cloud in 3D applications. In this section, we prove that the proposed encryption scheme maintains the stability of 3D objects. However, before we continue substantiating the stability of our encryption scheme, we firstly establish the notion of dimensional and spatial stability.

To elaborate the stability notion, let \( P^1, P^2, \ldots, P^n \) be points of the plain point cloud \( P \), and \( C^1, C^2, \ldots, C^n \) be points of the corresponding cipher point cloud \( C \). Given a point cloud, denote by \( G \) the barycentre of the point cloud, and by \( r \) the distance from the farthest vertex to \( G \). Any point cloud with \( (r, G) \) is encapsulated in a bounding sphere of radius \( r \) with center \( G \). Therefore, the bounding spheres for the plain point cloud \( P \) and cipher point cloud \( C \) are characterised by \( (r_P, G_P) \) and \( (r_C, G_C) \), respectively. To avoid visual inconsistencies and
therefore maintain the usability of the point clouds in virtual scenes, the encryption transformation should maintain the cipher-points within the bounding sphere of the plain point cloud; otherwise, the encrypted point cloud may overlap with other objects of the virtual world.

Spatial stability implies that the encrypted point cloud is placed at the same location as the original point cloud. In other words, the center of the bounding sphere of the encrypted point cloud is positioned inside the bounding sphere of the plain point cloud. More precisely, 

$$||G_C - G_P|| \leq r_P.$$  

Dimensional stability implies that the dimensional size of the encrypted point cloud is equal to or smaller than the dimensional size of the plain point cloud. In other words, the radius of the bounding sphere of the encrypted point cloud is less than the radius of the bounding sphere of the plain point cloud. More precisely, 

$$r_C \leq r_P - ||G_C - G_P||.$$  

The notion of dimensional and spatial stability implies that stability is inclusive, that is, the bounding sphere of the encrypted point cloud is included in the bounding sphere of the original point cloud. In addition, if the enclosing sphere of the encrypted point cloud is included in the bounding sphere of the original point cloud, one can easily infer that the requirements for the dimensional and spatial stability are satisfied. Therefore, the dimensional and spatial stability of the cipher point cloud is maintained if and only if the bounding sphere of the encrypted point cloud is included in the bounding sphere of the original point cloud, that is, inclusiveness is equivalent to dimensional and spatial stability.

We use the following lemmas to prove our claim (Theorem 4.1).

Lemma 4.1. Given $n$ random points $P_1, P_2, \cdots, P^n$ within a sphere of radius $r$ with center $G \in \mathbb{R}^3$, let $P'_1, P'_2, \cdots, P'^n$ be the result of permutation of dimensional coordinates of $P_1, P_2, \cdots, P^n$. For any $j (1 \leq j \leq n)$, $||P'_j - G|| \leq \sqrt{3}r$.

Proof. Permutation $\Pi : X \rightarrow X'$ is an injective and surjective mapping that assigns elements of a finite set $X$ of dimensional coordinates of $P_1, P_2, \cdots, P^n$ to itself (a finite set $X'$ of dimensional coordinates of $P'_1, P'_2, \cdots, P'^n$). For any $j (1 \leq j \leq n)$, $||P'^j - G|| \leq r$.

Hence, dimensional coordinates of $P'^j$ can have a value between $-r$ to $+r$. If permutation generates 3-tuples with maximal dimensional coordinates, such as $(\pm r, \pm r, \pm r)$, then the maximal distance of $P'^j$ from $G$ would be $\sqrt{3}r$. $lacksquare$

Lemma 4.2. Given three distinct points $P^1, P^2$ and $P^3$ in $\mathbb{R}^3$, let $P^3$ be the result of rotating $P^1$ about $P^2$ with an arbitrary angle. If $P^1$ and $P^2$ are arbitrary points inside a sphere of radius $r$ with center $G \in \mathbb{R}^3$, then $||P^3 - G|| \leq 3r$.

Proof. We start the proof by definition of a rotation. A rotation is a distance preserving transformation determined by the rotation center and Euler angles. To let the rotation sweep a larger area, the rotation distance $||P^1 - P^2||$ should be maximised. To this end, $P^1$ and $P^2$ must be positioned diametrically antipodal on the surface of the sphere. By fixing the rotation distance, the rotation maps $P^1$ to $P^3$, which is located on the surface of a larger sphere of radius $2r$ with
center $P^2$. Figure 4.2 depicts the maximal space that $P^3$ can appear. As shown in the figure, the maximal space is a bounded sphere of radius $2r$ with center $P^2$. By the triangle inequality, the following relationship is hence true:

$$
||P^3 - G|| \leq ||P^2 - G|| + ||P^3 - P^2|| = 3r.
$$

(4.19)

Figure 4.2: Maximal space that a random rotation can disperse a point.

**Theorem 4.1.** Given a plain point cloud $P = \{P^1, P^2, \ldots, P^n\}$ bounded by a sphere of radius $r$ with center $G$, the proposed encryption scheme maintains the dimensional and spatial stability of the corresponding cipher point cloud $C = \{C^1, C^2, \ldots, C^n\}$.

**Proof.** To prove this theorem, we need to measure dimensional and spatial deviations of the cipher point cloud from the plain point cloud. The proposed encryption algorithm is a combination of 3 consecutive procedures: a pseudorandom point generation process, a permutation process, and a geometric rotation process. Given $n$ plain-points $P^1, P^2, \ldots, P^n$ within a sphere of radius $r$ with center $G$, the first encryption step generates $n$ pseudorandom points within a sphere $S$ of radius $\sqrt{3}r$ with center $G$. According to Lemma 4.1, the permutation process generates a new set of points distributed in a sphere $S'$ of radius $3r$ with center $G$. By Lemma 4.2, random geometric rotations disperse the permuted points to a distance no more than thrice the radius of the sphere $S'$ from the center $G$. Figure 4.3 depicts the bounded spheres that every point may be mapped into after a random rotation. As shown in the figure, all spheres are located within a bigger sphere $S''$ whose radius is thrice the radius of the sphere $S'$ and center is $G$. Thus, the result of the geometric rotation process will be a set of points distributed within the sphere $S''$. To control the boundary of the encrypted point cloud and therefore to ensure the dimensional stability of the point cloud, a scaling factor is utilized in the geometric rotation process. The result of this analysis is the same for any number of encryption rounds. This proves that given
Chapter 4. 3D Object Encryption Which Maintains Dimensional and Spatial Stability

$n$ plain-points $P^1, P^2, \cdots, P^n$ within a sphere of radius $r$ with center $G$, their cipher-points $C^1, C^2, \cdots, C^n$ will be distributed in a sphere of the same radius with the same center. This meets the requirements for the dimensional and spatial stability, because the bounding sphere of the encrypted point cloud is included in the bounding sphere of the original point cloud. In other words, the bounding sphere of the encrypted point cloud is positioned inside the bounding sphere of the plain point cloud (spatial stability), and the size of the bounding sphere of encrypted point cloud is equal to or smaller than that of the plain point cloud (dimensional stability).

\[ \text{FIGURE 4.3: Boundary of the rotated point cloud.} \]

\[ \text{4.4 Cryptanalysis} \]

In this section, we evaluate the security of the proposed cipher using cryptanalytic methods. Theoretically, the security level of a cryptosystem is dependent on its key length. In literature, there are various recommendations for the appropriate key length of a particular encryption system \cite{44}, \cite{45}, \cite{46}. According to the guidelines released by NIST \cite{46}, 128 bit key length is an acceptable margin for designing secure symmetric-key encryption algorithms until 2030. The proposed cipher uses six seed points to initiate the Chebyshev map. Considering the double precision 64-bit IEEE 754 format \cite{154}, a 64-bit number is represented using 1 bit for the sign bit, 11 bits for the exponent width, and 52 bits for the fraction precision. In this format, the exponent is biased by adding 1023 (= 0b1111111111) before being stored. For any $i \ (1 \leq i \leq 6)$, $|k_i| \leq 1$. Hence, the most significant bit of biased exponent field always remains unchanged (equal to 0). However, the sign and the fraction parts use all their bits. Therefore, the key length of the proposed cipher is 378 (= $6 \times 63$) bits. Accordingly, the computational complexity of the exhaustive key search (ciphertext-only attack) is $2^{378}$. Table 4.1 compares the key space of the proposed cipher with a number of well-known 3D object encryption schemes, namely, the schemes by Shi et al. \cite{131} and Technicolor \cite{128}. Compared to Shi et al.’s and
Technicolor’s encryption schemes, the proposed cipher has a larger key space, which indicates a higher security level against brute-force attacks.

Table 4.1: Comparison of key space

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Key Space Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>Shi et al., 2006 [131]</td>
<td>$2^{128}$</td>
</tr>
<tr>
<td>Technicolor, 2013 [128]</td>
<td>$2^{192}$</td>
</tr>
<tr>
<td>Proposed</td>
<td>$2^{378}$</td>
</tr>
</tbody>
</table>

To break the cipher, an adversary has temporary access to the encryption and decryption machinery, and so is able to make queries of certain behaviours and observe the corresponding output without the knowledge of the key. This may help the adversary to determine, or at least partially determine, the secret key. In addition, for any $i (1 \leq i \leq 2n)$, deducing the expanded keystream, that is, \( \{k_{i}^{1}, k_{i}^{2}, k_{i}^{3}, k_{i}^{4}, k_{i}^{5}, k_{i}^{6}\} \), is totally equivalent to finding the secret key whenever different point clouds are encrypted using the same secret key.

Before we continue showing the security of our scheme, we firstly point out why simpler variants on the same idea are insecure, for example, if $E_{K}(P) = \Pi_{K}(P)$, such as encryption schemes of [130] and [128]. Choosing a permutation of large length size can exponentially increase the number of possible permutations $\#(\pi)$ of dimensional coordinates, that is,

$$\#(\pi) = (n!)^{3n}, \quad (4.20)$$

where $n$ is the number of 3D points. This exponential search space can make the statistical attacks cumbersome by increasing the size of a plain point cloud. However, permutation of a large number of coordinates, for instance, all of the dimensional coordinates at once, may not be an efficient solution because it can take a considerable amount of time to generate a cipher point cloud. In addition, permutation-only schemes are vulnerable to known/chosen plaintext attacks. In practical terms, only one pair of plain/cipher point clouds of $n$ points with non-repeated coordinates is sufficient to uniquely determine the permutation mapping $\Pi_{K}(\cdot)$ of length $3n$.

Another example is $E_{K}(P) = \text{Rot}_{K}(P)$. Given a pair of input/output point clouds $(P, C)$, for any $j (1 \leq j \leq n)$, $C^{j} = R^{j} \times [P^{j} - O^{j}] + O^{j}$. To determine $R^{j}_{3 \times 3}$ and $O^{j}_{3 \times 1}$, the adversary needs to solve the following system of equations:

$$
\begin{align*}
  c^{j}_{1} &= r^{j}_{11}(p^{j}_{1} - o^{j}_{1}) + r^{j}_{12}(p^{j}_{2} - o^{j}_{2}) + r^{j}_{13}(p^{j}_{3} - o^{j}_{3}), \\
  c^{j}_{2} &= r^{j}_{21}(p^{j}_{1} - o^{j}_{1}) + r^{j}_{22}(p^{j}_{2} - o^{j}_{2}) + r^{j}_{23}(p^{j}_{3} - o^{j}_{3}), \\
  c^{j}_{3} &= r^{j}_{31}(p^{j}_{1} - o^{j}_{1}) + r^{j}_{32}(p^{j}_{2} - o^{j}_{2}) + r^{j}_{33}(p^{j}_{3} - o^{j}_{3}).
\end{align*}
\quad (4.21)$$
For any \( j \) \((1 \leq j \leq n)\), the adversary therefore requires only 4 pairs of plain/cipher point clouds to construct a system of 12 nonlinear equations and uniquely determine the unknown matrices, which are \( R_j^{3 \times 3} \) and \( O_j^{3 \times 1} \).

Another variant is a one-round permutation-rotation structure, that is, \( E_K(P) = R_{ot_K}(\Pi_K(P, O)) \). To break this variant, for any \( i \) \((1 \leq i \leq 3n)\) and \( j \) \((1 \leq j \leq n)\), the adversary needs to determine the permutation keystream \( \pi_i \) and the unknown matrices, which are \( R_j^{3 \times 3} \) and \( O_j^{3 \times 1} \). Since permutation of dimensional coordinates is performed within the groups of 8 plain-points and their corresponding pseudorandom points, the adversary needs to observe the 8 consecutive geometric rotations, that is, 8 systems of equations such as equation 4.21, at once. Due to the permutation process, for \( l \) \((1 \leq l \leq 8)\), there are 48! different possible arrangements for the dimensional coordinates of \( P_l \) and \( O_l \). Hence, there are 48! different possibilities for every 8 systems of equations. As explained above, for any \( j \) \((1 \leq j \leq n)\), only 4 pairs of plain-points/cipher-points is sufficient to determine \( R_j^{3 \times 3} \) and \( O_j^{3 \times 1} \). Therefore, the adversary can attack this variant by a known-plaintext attack using 4 pairs of plain/cipher point clouds with the computational complexity of 48! = \( 2^{202.9} \) encryption. The complexity of this attack is much less than the exhaustive key search, that is 378. To reduce the complexity of this attack, a chosen-plaintext attack can be employed to reduce the search space for determining the permutation keystream. To this end, the adversary can exchange the indices of at least two points of the plain point cloud (point reordering attack [149]) and observe the changes which occur at the cipher point cloud. For any \( m \) \((1 \leq m < \lfloor \frac{n}{8} \rfloor)\), \( \Pi_{\{\pi_i: 48(m-1)+48 \leq i \leq 48(m-1)+1\}} : X_m \rightarrow X'_m \).

In other words, every permutation is performed between dimensional coordinates of 16 points (8 consecutive plain-points and 8 consecutive pseudorandom points) in \( X_m \). Exchanging the indices (positions) of at least two points of the plain point cloud can, at most, affect the coordinates of 6 points of \( X_m \). If these 6 affected points hold different indices, then, in the worst case, every 6 out of 8 points (\( 75\% \)) of the cipher point cloud would be affected by geometric rotation. To determine the permutation mapping of whole elements in every \( X_m \), the adversary can repeat the point index shifting 8 times. This reduces the search space of permutation mapping from 48! to 36!. As explained above, only 4 pairs of plain/cipher point clouds are sufficient to determine the geometric rotation. Therefore, this variant is broken by a chosen-plaintext attack with 12 pairs of plain/cipher point clouds and computational complexity of 36! = \( 2^{138} \) encryption. This attack is more efficient than the known-plaintext attack with complexity 202.9.

Now we analyse the security of the proposed cipher (a two-round permutation-rotation scheme), that is, \( E_K(P) = R_{ot_K}(\Pi_K(P, O))^2 \). To break the cipher, for any \( i \) \((1 \leq i \leq 3n)\) and \( j \) \((1 \leq j \leq n)\), the adversary could employ a similar attack procedure explained for the one-round variant to determine the permutation keystream \( \pi_i \) and the unknown matrices, which are \( R_j^{3 \times 3} \) and \( O_j^{3 \times 1} \). By observing every block of 8 consecutive plain-points separately, there are 8 consecutive geometric rotations of plain-points about pseudorandom points in each block.
Mathematically, this is equivalent to 8 systems of 12 non-linear equations. Due to the permutation procedure, these 8 systems of equations can have $48!$ different arrangements in each round. Therefore, there are $(48!)^2$ different arrangements for every 8 systems of equations after 2 rounds. Also, for any $j (1 \leq j \leq n)$, 4 unknown matrices, which are $R^j_{3 \times 3}$ and $O^j_{3 \times 1}$ in the first round, and $R^{ij}_{3 \times 3}$ and $O^{ij}_{3 \times 1}$ in the second round, need to be determined. As explained above, the adversary needs at least 8 pairs of known plain/cipher point clouds to determine the geometric rotation in both rounds. Therefore, the data complexity of the known-plaintext attack is 8 pairs of plain/cipher point clouds and its computational complexity is $(48!)^2 = 2^{405.8}$ encryption. This attack strategy (known-plaintext attack) is less efficient than the exhaustive key search, and therefore, it is not feasible on the two-round permutation-rotation encryption scheme. To reduce the attack complexity, the adversary may use a point reordering attack (chosen-plaintext attack) to reduce the search space for determining the permutation keystream. However, this approach does not work because exchanging the indices of two points of the plain point cloud can change the coordinates of 6 points of $X_m$ by the first round and by the second round, it can change the coordinates of all points of $X_m$. Hence, the point reordering attack is not feasible on the proposed cipher and compared to the known-plaintext attack, it cannot reduce the search space for determining the permutation keystream. We thus conjecture that the proposed encryption scheme is secure from known-plaintext attacks and chosen-plaintext attacks.

### 4.5 Statistical Analysis

Statistical analysis evaluates the statistical properties of an output stream of a cipher, independent of knowledge of the cipher structure. The failure in statistical tests indicates a bias in the cipher output, and hence, shows that it can be predicted from input. While having good statistical properties alone cannot guarantee the cryptographic security, statistical analyses are a compliment to cryptanalysis. In this section, we perform a number of tests, including a similarity analysis, a plaintext sensitivity analysis, a key sensitivity analysis, and a randomness analysis, to evaluate the statistical properties of the proposed 3D object encryption scheme. The evaluation consisted of both theoretical analysis and practical well-known experimentation. To perform the tests, we have chosen a number of plain point clouds from [1]. We also used different sets of keys to perform the tests: $K = (0, 0.2, -0.5, 0.9, -0.8, 0.1)$ as the original key and $K' = (0.1, 0.2, -0.5, 0.9, -0.8, 0.1)$ as the slightly modified key.

#### 4.5.1 Similarity Analysis

The ability of the adversary to have access to a number of plaintext and ciphertext pairs can help them to perform a similarity analysis between the plaintext and ciphertext, and therefore, obtain additional information about the encryption mapping. This may help the adversary to
learn about the plaintext using the ciphertext information, without the knowledge of the secret key. To ensure the security of a 3D content encryption system, cipher objects must leak no information regarding the geometric structure of plain objects.

A straightforward similarity analysis is the pairwise Euclidean distance between the points of the point clouds under study. This measure provides the point-wise similarity information of the point clouds. Given a set $P$ of $n$ points $P^1, P^2, \ldots, P^n$, for any $i$ ($1 \leq i \leq n$) and $j$ ($1 \leq j \leq n$), the Euclidean distance (2-norm distance) between $P^i$ and $P^j$ is defined as follows [155]:

$$d(P^i, P^j) = \left( \sum_{m=1}^{3} \left| p^i_m - p^j_m \right|^2 \right)^{\frac{1}{2}}. \tag{4.22}$$

For any $i$ ($1 \leq i \leq n$), the pairwise Euclidean distance $d_p$ between the corresponding points of two point clouds $P$ and $C$ is

$$d_p^i = d(P^i, C^i). \tag{4.23}$$

Clearly, if this difference is non-zero, then it shows that the encryption procedure scatters the plain-points to different locations. The pairwise Euclidean distance metric is a one-dimensional method and cannot provide any information about the rigid similarity of surfaces (objects). Using such a one-dimensional method for a multi-attribute analysis can lead to making wrong conclusions. Therefore, in addition to the pairwise Euclidean distance, we use the Hausdorff distance [156] to detect similar surfaces with rigid isometries, such as shifting and rotation.

Let $P$ and $Q$ be two non-empty subsets representing two rigid surfaces (objects). For any $i$ ($1 \leq i \leq \#P$) and $j$ ($1 \leq j \leq \#Q$), the Hausdorff distance is defined as follows [156]:

$$d_H(P, Q) = \max \left\{ \sup_{P^i \in P} \inf_{Q^j \in Q} d(P^i, Q^j), \sup_{Q^j \in Q} \inf_{P^i \in P} d(P^i, Q^j) \right\}, \tag{4.24}$$

where sup represents the supremum and inf the infimum. The Hausdorff distance determines whether $P$ and $Q$ represent the same rigid surface (object) or not. To measure the dissimilarity between the plain and cipher point clouds, a number of similarity experiments were performed. Figure 4.4 depicts the results of a similarity analysis for a circular loop of 10000 points with radius 2 and its cipher point cloud. A visual observation indicates that the encryption process pseudo-randomly scatters plain-points into a limited space. It also shows that the encryption result of a circular, ring-shaped point cloud has a different shape (approximately ball-shaped). This observation verifies that the visual information (meaningful pattern) of the plain point cloud is completely damaged and a noisy aspect is observed. The result obtained from the pairwise Euclidean distance between the corresponding points and the Hausdorff distance confirms the visual analysis. As shown in Figure 4.4c, all points of the plain point cloud are displaced from their original location. For a better comparison, a sample of the plain point cloud with size 350 is depicted in a 2D plot (Figure 4.4d) along with its corresponding cipher point cloud. Figure
Chapter 4. 3D Object Encryption Which Maintains Dimensional and Spatial Stability

4.4d shows that the Hausdorff distance between the plaintext and ciphertext is the maximum minimum distances between the plaintext and ciphertext. The Hausdorff distance being non-zero shows that the plaintext and ciphertext clouds are dissimilar. Figure 4.4e depicts the heat map of the distance matrix, where entry $(n, m)$ is the distance of the $n$-th point in the plain point cloud from the $m$-th point in the cipher point cloud.

The visual summary given by the heat map suggests dissimilarity of the point clouds under study. In comparison with previous dimension and space preserving schemes, such as [130], [133], and [128], our proposal has a better performance with regard to similarity analysis. For instance, given a point cloud of $n$ vertices with zero coordinates, previous schemes ([130], [133],...
Chapter 4. 3D Object Encryption Which Maintains Dimensional and Spatial Stability

and \([128]\)) render the same output while the proposed scheme generates a completely different point cloud.

4.5.2 Plaintext Sensitivity Analysis

In general, a desirable property for an encrypted point cloud is being sensitive to minor alterations in the plain point cloud, for instance, modifying only the position of one point. To study the relationship between the plaintext and ciphertext, the adversary may slightly change the position of one point in the point cloud and observe the changes in the encrypted point cloud. By this method, the meaningful relationship between the original point cloud and the encrypted point cloud can be found, which further facilitates in determining the secret key. If a small change in the position of one point in the original point cloud changes the position of a significant number of points in the encrypted point cloud, then the differential attack becomes practically infeasible. In the proposed cipher, each point is rotated randomly with respect to a random rotation reference which is calculated using the point cloud’s center of mass. A tiny change in the point cloud’s center of mass can affect the result of all random rotations, and therefore, it can result in a completely different cipher point cloud. For instance, if the cryptanalyst changes the position of one point, then the point cloud’s center of mass will be changed. This therefore changes the position of other points in the encrypted point cloud. This shows that the proposed cipher is sensitive to changes to the plaintext, and it is robust to differential cryptanalysis. This analysis is also confirmed by our simulation result shown in Figure 4.5. In this experiment, we have chosen Michael11, which contains 52560 points, as the plain point cloud and encrypted it using the original key. We also modified the plain point cloud by displacing 1 point by a distance 0.1\% of radius \(r\) of the bounding sphere, and encrypted it using the original key. To observe the changes in the encrypted point clouds, scatter plots of the encrypted point clouds are depicted in Figures 4.5.b and 4.5.d. For a better comparison, the pairwise Euclidean distances between the corresponding points of two plain point clouds and two encrypted point clouds are calculated and results are depicted in Figure 4.6. It can be observed that a small change in the position of even one point in the original point cloud will result in a significant change in the location of cipher-points. In addition to the pairwise Euclidean distance analysis, we also studied the impact of increasing the number of displaced points and the ratio of displacement on the dissimilarity between surfaces of the cipher point clouds. To this end, we plotted multiple Hausdorff distance curves (Figure 4.7) by varying the ratio of modified points in the plain point cloud, that is, 1 point, 1\% and 2\% of points moved by a distance between 0.1\% and 7\% of radius \(r\) of the cipher object’s bounding sphere. The result of this analysis shows that the shape of the cipher point cloud alters by changing the number of altered points and the ratio of spatial displacement. This indicates that the plaintext sensitivity of the proposed scheme is a function of both the ratio of points changed in the plain point cloud, and the ratio of displacement. Hence, any plaintext alteration within the plain point cloud not only changes the cipher-points but also changes the
shape of the cipher point cloud. This indicates the robustness of our scheme to any differential analysis.

![Diagram](image)

**Figure 4.5:** Plaintext sensitivity test result: (a) original plain point cloud, (b) encrypted point cloud from the original plaintext, (c) slightly changed plain point cloud by displacing 1 point by a distance 0.1% of radius $r$ of the bounding sphere, and (d) encrypted point cloud from the slightly changed plaintext.

In comparison with previous dimension and space preserving schemes, such as [130], [133], and [128], our proposal is more sensitive with regard to small plaintext alterations. For instance, given Michael11 as the input point cloud, changing the location of only 1 point will only change one cipher-point in Pan et al.’s encryption scheme [130], and at most 3 cipher-points in Technicolor’s encryption scheme [128]. It also has no impact on the rendered bounding box in Phelps’s encryption scheme [133]. In addition, such modifications do not change the shape of cipher point clouds by the encryption scheme of [130], [133], and [128], and create similar cipher point clouds with zero Hausdorff distances. As explained in the cryptanalysis section (see Section 4.4), this can help an adversary to easily track the alterations and deduce the encryption mapping.
Figure 4.6: Pairwise Euclidean distance between the corresponding points of (a) plain point clouds and (b) encrypted point clouds.

Figure 4.7: Log-log plot of the Hausdorff distance against the ratio of the spatial displacement.

4.5.3 Key Sensitivity Analysis

A 3D content encryption scheme should be sensitive to changes to the secret key. In other words, a change in a single bit of the secret key should produce a completely different cipher point cloud. To test the key sensitivity of the proposed algorithm, a number of point clouds were encrypted using the original secret key and a slightly modified secret key. As it was not easy to compare the encrypted point clouds by simply observing them, the pairwise Euclidean distance between the corresponding points of two encrypted point clouds were calculated. Figure 4.8 shows the result of key sensitivity analysis for Centaur5 [1]. It is observed that two encrypted point clouds with a slightly different key are quite different. This indicates the high sensitivity of the proposed method to changes of the key. In comparison with previous encryption schemes, such as [130] and [128], the proposed encryption scheme is more sensitive to the changes of the
secret key. For instance, given a point cloud of \( n \) vertices with zero coordinates, changing the secret key creates the same point cloud by the encryption scheme of [130] and [128], while the proposed cipher generates a completely different cipher point cloud.

\[
\text{FIGURE 4.8: Key sensitivity test result: (a) plain point cloud } \text{Centaur5} [1], \text{ (b) encrypted point clouds using the original and slightly modified secret keys, and (c) Euclidean distance between points of two encrypted point clouds.}
\]
4.5.4 Spatial Randomness

To ensure the security of a 3D encryption system, the 3D cipher must have good probabilistic properties, one of which is random distribution. More specifically, the points of a plain point cloud are required to be dispersed as randomly as possible. This can annihilate any distinguishable patterns or shapes within the original object. This is desirable because the existence of any distinguishable pattern or relationship among the points of a cipher point cloud may lead to data leakage, which may help the adversary partially discover the plain object and hence break the cipher. Any competent adversary would attempt to acquire any knowledge from the point cloud, and therefore, would recognize any hidden pattern to then enable successful reconstruction of the surface. A primary analysis that an adversary may take into consideration is measuring the distance between each pair of points within the ciphertext cloud. The purpose of such analysis would be to cluster any existing pattern inside the ciphertext cloud, such that points in the same cluster have a small distance from one another, while points in different clusters are at a large distance from one another. This may allow partial reconstruction of the surface from an unorganised point cloud. To resist such analysis, the cipher point cloud must be distributed as randomly as possible. Spatial randomness of cipher-points suggests that it would be hard to find any clusters among so many pairs that are all at approximately the same distance.

In a good point cloud encryption scheme, cipher-points are equally likely to occur at any location and the position of any cipher-point is not affected by the position of any other point. In other words, cipher-points occur within a given study volume with no apparent ordering of the distribution. Therefore, in comparison with plain-points, cipher-points are specific point events. This indicates that a good point cloud encryption scheme is synonymous with a homogenous three dimensional Poisson process. In this process, the expected number of points per unit volume (intensity) is

$$\rho = \bar{\rho} = \rho(v),$$  \hspace{1cm} (4.25)

where $v$ is the spatial volume of the study element, which is a sphere of radius $r$. Hence, a suitable estimator for $\rho$ is

$$\hat{p} = \frac{\int_V \rho(v) \, dv}{\int_V dv}. \hspace{1cm} (4.26)$$

For a random distribution of sample points in a 3D space, the probability that a sample volume of a specific size (event $X$) will contain exactly $x$ points is represented by the Poisson’s exponential function, that is,

$$Pr(X = x) = \frac{\lambda^x e^{-\lambda}}{x!}, \hspace{1cm} (4.27)$$

where $Pr(\cdot)$ denotes the probability function and $\lambda$ is the mean number of points per volume. Let the sample volume be a sphere of radius $r$. Under this assumption, the mean number of
points is
\[ \lambda = \rho \left( \frac{4}{3} \pi r^3 \right), \]  
(4.28)
where \( \rho \) is the mean density of the distribution. In this case, the probability of finding exactly \( x \) points in the sample area will be
\[ Pr(X = x) = \frac{\rho^x}{x!} e^{-\rho \left( \frac{4}{3} \pi r^3 \right)}. \]  
(4.29)

The probability that there are no points within a distance \( r \) of a chosen point in the area under study, is equal to the observation of 0 from the Poisson distribution which is given by
\[ Pr(X = 0) = e^{-\rho \left( \frac{4}{3} \pi r^3 \right)}. \]  
(4.30)

Accordingly, the probability that the area under study contains one or more points within the distance \( r \) of a chosen point will be
\[ Pr(X \geq 1) = 1 - e^{-\rho \left( \frac{4}{3} \pi r^3 \right)}. \]  
(4.31)

This expression also represents the proportion of distances to the nearest neighbour. Thus, the probability distribution \( f(r) \) of the distance \( r \) of a randomly selected point to its nearest neighbour is obtained by differentiating the proportion of distances to the nearest neighbour.
\[ f(r) = \frac{\rho}{k} \left( \frac{4}{3} \pi r^2 \right) e^{-\rho \left( \frac{4}{3} \pi r^3 \right)}, \]  
(4.32)

The first moment of \( r \), namely mean of \( r \) or \( E(r) \), can be obtained as follows:
\[ E(r) = \int_0^{\infty} r \cdot \frac{\rho}{k} \left( \frac{4}{3} \pi r^2 \right) e^{-\rho \left( \frac{4}{3} \pi r^3 \right)} dr = \frac{1}{9} \left( \frac{2 \left( \frac{4}{3} \right) 3 \left( \frac{4}{3} \right) \pi \left( \frac{4}{3} \right)}{\left( \frac{\rho}{k} \right) \left( \frac{4}{3} \right) \gamma \left( \frac{4}{3} \right)} \right) = 0.5540 \left( \frac{\rho}{k} \right)^{-\frac{1}{3}}. \]  
(4.33)

The second moment of \( r \), namely variance of \( r \) or \( E(r^2) \), can be obtained as follows:
\[ E(r^2) = \int_0^{\infty} r^2 \cdot \frac{\rho}{k} \left( \frac{4}{3} \pi r^2 \right) e^{-\rho \left( \frac{4}{3} \pi r^3 \right)} dr = \frac{1}{6} \left( \frac{6 \left( \frac{4}{3} \right) \gamma \left( \frac{4}{3} \right)}{\left( \frac{4}{3} \right) \left( \frac{4}{3} \right) \gamma \left( \frac{4}{3} \right)} \right) = 0.3474 \left( \frac{\rho}{k} \right)^{-\frac{2}{3}}. \]  
(4.34)

As explained above, the mean and variance of the distance \( r \) of a randomly selected point to its nearest neighbour are 0.5540 \((\rho)^{-\frac{1}{3}}\) and 0.3474 \((\rho)^{-\frac{2}{3}}\), respectively, where \( \rho \) is the expected number of points per unit volume (intensity). For further information about the calculation of the model parameter values, that is, the mean and variance of \( r \), please see [157] and [158].

To investigate the robustness of the proposed encryption scheme to surface reconstruction attacks, we evaluate the spatial randomness of cipher-points using the nearest neighbour method [159]. To this end, we use the Euclidean distance between the nearest neighbours as our statistic.
to study the spatial distribution of the cipher-point. For any \(1 \leq i \leq n\) and \(1 \leq j \leq n\), the nearest neighbour distance of point \(P^i\), that is, the distance of point \(P^i\) to its nearest neighbour point in \(P\), is defined as follows:

\[
d_i = \min \left\{ d \left( P^i, P^j \right) \right\},
\]

(4.35)

where \(P^i\) and \(P^j\) \(\in P\), and \(i \neq j\). Taking the nearest neighbour distance as the statistic would violate the independence assumption of events in a Poisson process. One can easily observe that for any \(1 \leq i \leq n\) and \(1 \leq j \leq n\), where \(i \neq j\), if \(P^i\) and \(P^j\) are mutual nearest neighbours, then \(d_i = d_j\). Thus, \(P^i\) and \(P^j\) are clearly not independent. To resolve this problem, we consider the mean of nearest neighbour distances in a randomly selected subset of a point cloud, as follows:

\[
\overline{d}_j = \frac{1}{j} \sum_{i=1}^{j} d_i.
\]

(4.36)

According to the central limit theorem (CLT), under the spatial randomness hypothesis (null hypothesis) and for a sufficiently large sample point cloud with independently and identically distributed \(iid\) points, the mean of nearest neighbour distances \(\overline{d}_j\) must be approximately normally distributed, with the following mean and variance [160]:

\[
\overline{d}_j \sim N \left( 0.5540 (\rho)^{-\frac{1}{3}}, \frac{0.3474}{j} (\rho)^{-\frac{2}{3}} \right).
\]

(4.37)

To construct the spatial randomness test, the sample mean \(\overline{d}_j\) needs to be standardized. Under the spatial randomness hypothesis, the standardized sample mean, that is, \(Z_j \sim N(0,1)\), is calculated as follows:

\[
Z_j = \frac{\overline{d}_j - E(\overline{d}_j)}{\sigma(\overline{d}_j)} = \frac{\overline{d}_j - 0.5540 (\rho)^{-\frac{1}{3}}}{\sqrt{\frac{0.3474}{j} (\rho)^{-\frac{2}{3}}}}.
\]

(4.38)

In the standard normal distribution,

\[
Z_j \sim N(0,1), \quad Pr \left( Z \geq z_\alpha \right) = \alpha.
\]

(4.39)

Hence, according to the standardisation procedure,

\[
Pr \left( |Z_j| \geq z_\frac{\alpha}{2} \right) = Pr \left[ \left( Z_j \leq -z_\frac{\alpha}{2} \right) \text{ or } \left( z_\frac{\alpha}{2} \leq Z_j \right) \right] = \alpha.
\]

(4.40)

Equation 4.40 shows the significance of departure from random expectation. If the null hypothesis is valid, then \(Z_j\) should be a sample from \(N(0,1)\). The null hypothesis is rejected if and
only if $|Z_j| \geq z_{\alpha/2}$. To interpret the test results, the $P$-value can be reported as follows:

$$P\text{-value} = Pr\left(|Z| \geq z_{\alpha/2}\right) = 2\Phi\left(-|z_{\alpha/2}|\right),$$

(4.41)

where $\Phi(\cdot)$ denotes the cumulative distribution function. If $P$-value $< \alpha$, then the spatial randomness hypothesis of the cipher point cloud, that is, the null hypothesis, is rejected. To check the presence of any pattern in the point distribution of the cipher-text point clouds, we have applied the spatial randomness test on the ciphertext result of a number of 3D objects chosen from [1]. According to test results, for all cipher-text point clouds $|Z_j| \leq z_{0.025} = 1.96$ and $P$-value $> 0.05$, which shows that with the standard significance level of $\alpha = 0.05$ the spatial randomness is not rejected and the encryption scheme under study passes the statistical test. This indicates a good statistical property of the encryption algorithm which can successfully dissipate any meaningful relationship between the points of the plain point cloud. Figure 4.9 shows the result of the spatial randomness test on Michael1's cipher point cloud. To draw the histogram of mean nearest neighbour distances, 5000 samples of size $j = 100$ were selected, and the corresponding $Z$-values were calculated. As shown in the figure, the results of this simulated sampling scheme yield a distribution of $Z$-values that is approximately normal. The mean of this distribution is $-1.5258$ and its $P$-value is $0.12705$. This shows that no clusters can be identified within the cipher point cloud. Results of our analysis for the permutation-only encryption schemes, such as [130] and [128], show that such schemes fail the spatial randomness test and they do not disperse the plain-points randomly into the bounded space.
4.5.5 Normal Vector Analysis

To achieve proper lighting of curved surfaces, vertex normal vectors need to be computed at certain points in a 3D graphics pipeline. Vertex normals indicate the orientation of the surface at each vertex and are used to produce smoother rendering results than flat shading. Therefore, to facilitate the visual interpretations, a normal vector is also included in some point cloud formats [161], such as PLY, STL and OBJ. This can reduce the computational time by avoiding the normals recreation. The proposed encryption scheme ignores the vertex normal information and only protects the vertex information. In this section, the consequences of ignoring the vertex normal information on the security of the proposed cipher are described.

A surface normal for a face can be calculated by taking the vector cross product of two edges of that face, and a vertex normal is the normalized average of the surface normals of the faces that contain that vertex. A primary attack to obtain the vertex information may be to decompose each vertex normal vector into the adjacent face normals, and then, decompose each face normal into vertex coordinates. Forming this system of equations may not be helpful to the adversary because it provides no unique solution for vertices.

Another attack to obtain the vertex information may be to recalculate the vertex normals of the cipher point cloud, and then, compute a displacement map between the plaintext and cipher-text normal vectors. This map can be used to displace a refined version of an approximate 3D geometry of the plain point cloud. Koller et al. have shown that their protection method is vulnerable to this type of attack [11] and a potential adversary is able to successfully reconstruct an approximate 3D geometry of the plain point cloud. However, this attack is not effective on the proposed cipher and it cannot uniquely determine the cipher point cloud because unlike Koller et al.’s protection method, which ignores the plain-points, the proposed encryption scheme scatters the plain-points into a bounded space. To exploit the displacement mapping, the adversary needs to apply it on an approximate 3D geometry of the plain point cloud. Results of similarity analysis and spatial randomness testing indicate that the plain and cipher point clouds are totally different. Therefore, it is difficult to obtain an approximate 3D geometry of the plain point cloud by the ciphertext vertex information. In addition, results of the plaintext sensitivity analysis indicate that the cipher point cloud is very sensitive to changes of the plaintext vertices. This shows that different plaintext/ciphertext pairs have different normal displacement mapping. Hence, given a normal displacement mapping for a known plaintext/ciphertext pair, it cannot be applied on any other ciphertexts to obtain the corresponding approximate plaintext 3D model. Figure 4.10 depicts results of surface reconstruction using the normal vector analysis. In this experiment, Cat0 is firstly encrypted using the original secret key. Figure 4.10.a shows Cat0’s 3D mesh. Figure 4.10.b shows the reconstructed surface of the unorganised cipher point cloud using Hoppe’s method [162]. It is clearly shown that the resulting surface is significantly different from that of the plaintext’s 3D mesh. To refine the reconstructed surface, we recalculated
the ciphertexts’ vertex normal vectors and computed their difference from the plaintexts’ vertex normal vectors. Using this information, we translated new faces and displaced the cipher mesh with a refined 3D mesh. As shown in this figure, the refined mesh is significantly different from the plaintext’s 3D mesh. Figure 4.10c shows the refined 3D mesh. This experiment shows that a competent adversary would not be able to reconstruct an acceptable 3D mesh from a cipher point cloud using the normal vector analysis. This resistance is mainly due to the random dispersion of the cipher point cloud.

### 4.6 Performance Analysis

In addition to security analysis, the encryption performance of 3D content is also an important factor to consider, especially for real-time applications which require a high level of efficiency. Generally, encryption performance hinges upon the structure of the central processing unit, programming language, memory size and the operating system. Since the proposed cipher is a symmetric algorithm, the encryption and decryption performance are the same. To evaluate the performance of the proposed cipher, the encryption scheme was implemented using an unoptimized MATLAB code on a machine with Intel Core i5 2.5 GHz processor and 4 GB of installed memory running under Windows 7. In addition, to having an accurate benchmark result, each timing test was executed 10 times and the average time was reported. The results of encryption time for encrypting 100 point clouds of various sizes ($10^2$ to $10^5$ points) are presented in Figure 4.11. The encryption time is then used to calculate the throughput (encryption speed) of the proposed algorithm, that is, the point cloud size (number of points) divided by the encryption time. This analysis indicates that on average, the proposed cipher encrypts roughly 12196 points per second. Considering the double precision format, each point is stored using $3 	imes 64$ bits; hence, the average throughput is 292.704 kilobytes per second.

The computational complexity of the proposed cipher is the summation of complexities of the key scheduling algorithm and its three components: a pseudorandom point generation, a
permutation, and a geometric rotation. The computational complexity of the key scheduling algorithm mainly depends on the implementation of the Chebyshev map. As determined by Brent and Zimmermann [163], the computational complexity of cosine function and its inverse can be calculated from log function and is \( O(M(a) \log(a)) \), where \( M(a) \) is the cost of multiplication and \( a \) is the number of digits of precision. As \( a \) is determined by the computing system, both \( a \) and \( M(a) \) are fixed and independent of the input point cloud. Thus, they are constant terms. Accordingly, the computational cost of generating the chaotic keystream for two round encryption is \( 12M(a)\log(a) \cdot n \). The computational complexity of a pseudorandom point generation is a constant term. Hence, the computational complexity of generating \( 2n \) points is \( d \cdot 2n \), where \( d \) is a constant. The computational complexity of each permutation mapping and each geometric rotation is a constant term. As explained in Algorithm 1, the \( n \) input points are partitioned into \( \lfloor \frac{n}{8} \rfloor \) subsets of points. The permutation mapping is firstly performed within every set of eight points and their corresponding pseudorandom points, and then all points of each set are geometrically rotated about their corresponding pseudorandom points. In each subset, the computational complexity of local permutations and geometric rotations is \( O(1) \). Therefore, the computational complexity of the one-round permutation-rotation process is \( O(n) \). Accordingly, the computational complexity of the encryption algorithm is \( O(n) \). This complexity estimation is confirmed by our simulation result shown in Figure 4.11. 3D content protection schemes in [11], [130], [131], and [128], have the same computational complexity as the proposed cipher, that is, \( O(n) \), where \( n \) is the number of points. However, as discussed earlier (Section 4.4), the scheme in [11] is not applicable to the point cloud representation, and schemes in [130] and [128] cannot ensure the security of point cloud vertices.

### 4.7 Conclusion

In this chapter, to overcome the limitations of the current techniques in addressing the confidentiality requirement of 3D objects, we proposed a technical solution for encrypting 3D objects. The proposed cipher, which is based on a series of random permutations and rotations, is compatible with standard file formats and maintains the semantic requirements of 3D objects, including the dimensional and spatial stability. Since the inverse of permutation and rotation matrices is their transpose, the implementation of the decryption scheme is very efficient. The cipher displaces the plain-points, and thus, deforms the geometry of the 3D object. This deformation preserves the dimensional and spatial stability of the original object. The security of the proposed cipher was convincingly verified by the evidence obtained from cryptanalytic methods and statistical analyses. The result of rigorous cryptanalysis indicated that the proposed encryption scheme is secure against known-plaintext attacks and chosen-plaintext attacks. The performed statistical tests included similarity analysis, plaintext sensitivity analysis, key sensitivity analysis, and spatial randomness analysis. The result of similarity analysis indicated that the plaintext and ciphertext clouds are not only dissimilar but also have dissimilar surfaces. It
was shown that the proposed cipher is sensitive to changes of the plaintext and key, and is robust to differential cryptanalysis. The result of the spatial randomness test indicated that there is no appearance of homogeneous zones in the spatial distribution of the cipher point cloud. This shows that the proposed cipher disperses the plain-points randomly into the bounded space. Results of statistical analyses indicated that the proposed encryption scheme is robust against surface reconstruction attacks. In addition to security analysis, the performance of the proposed cipher was tested, and its efficiency for 3D object encryption was validated. Finally, a comparison with existing protection methods showed that the proposed cipher is more effective and has better security despite having the same level of computational complexity.

In the next chapter, we propose a technical solution to prevent the disclosure of 3D surface geometry from the texture leakage using a texture encryption scheme that is lightweight and satisfies the security requirement. The proposed solution meets the constraints imposed by the structure of texture images, such as large data volume, and the application requirements, such as real-time performance, complexity, and the security level.
Texture Encryption and Maintaining the Secrecy of Protected 3D Objects

5.1 Introduction

Over the past decade, there has been a substantial rise in usage and distribution of 3D content in various industries. It is anticipated that the market for 3D mapping and 3D modelling will reach $7.7 billion by 2018 at a CAGR of 47.9% from 2013 to 2018 [164]. The growing applicability of 3D content and its potential revenue suggest the necessity for protecting such assets using encryption. Over the past four decades, a large number of encryption schemes have been proposed, among which DES [50] and AES [51] have been standardized and adopted worldwide. However, the problem of 3D content encryption is beyond the simple application of established and well-known encryption algorithms. This is primarily due to the constraints imposed by 3D content’s structure and application requirements, such as content usability, format compliance, real-time performance, complexity, and the security level. To address these concerns, several attempts have been made to develop robust encryption schemes for 3D content [11], [133], [128], [134], [12]. However, all these efforts are mainly focused on the encryption of 3D models rather than texture images. In this chapter, we propose a technical solution for the problem of texture image encryption which not only maintains the confidentiality of texture images but also maintains the security of protected 3D models from surface reconstruction attacks using the data provided by the texture images.
Texture images are fundamental drawing primitives that add realism to computer graphics by improving surface details. Notionally, a texture image is a 2D image that is wrapped onto the geometry of a 3D model, to give the illusion of a specified pattern to the complex object. Texture images contain intelligible information due to the strong correlation among adjacent elements. As each element is assigned to a particular vertex, texture patterns provide strong cues to the surface orientation, curvature and 3D surface geometry of the complex object. Therefore, there is a strong correlation between the geodesic distance between pairs of points on the surface and the distance between corresponding pairs of points in the texture image. One security issue is in the information that this relationship provides about the 3D geometry. Another issue is the texture leakage itself which may lead to the disclosure of the 3D surface geometry. It is therefore necessary to confuse this relationship by encrypting the texture image.

Texture encryption is a subclass of image encryption in which maintaining the real-time rendering performance, and preventing the partial disclosure of the 3D surface geometry by the texture pattern, are principally important in addition to providing confidentiality for texture images. These requirements may not be an issue in many image or video applications, but they are vital for most 3D applications. To meet these requirements, it is more important to obfuscate the coarse pattern rather than the detail of the texture image. This can reduce the capability of a competent adversary to reconstruct 3D objects exploiting the texture images. It is also necessary to keep encryption complexity as low as possible to save resources, such as computation, memory and bandwidth. One potential solution to the problem of texture encryption is in the use of a lightweight encryption scheme with a high level of security, tailored for maintaining real-time performances. Using this idea, this chapter proposes a novel texture encryption scheme that satisfies the need for both lightweightedness and security. The proposed cipher uses Salsa20/12 as its core encryption primitive. Salsa20 is one of the finalists of the eSTREAM project and is constructed by a simple and scalable design, which is appropriate for software implementations. Although Salsa20 has not received its deserved attention compared to AES, it has considerable potential for being used in multimedia applications, in which high-speed encryption is required. It is shown that in comparison with full encryption methods using the 128-bit AES, the proposed texture encryption method provides a comparable level of security but with much faster performance. Also, compared to selective encryption methods, in which only a subset of the input bitstream is encrypted using the 128-bit AES, the proposed texture encryption method is much faster and more secure because it protects the entire input bitstream. Furthermore, it is shown that Salsa Dance conceals the shape and boundaries of underlying 3D objects, while the full and selective encryption using AES cannot protect such information.

This chapter proposes the first commercially viable technical solution for the confidentiality problem of texture images, based on the findings of the literature review in Chapter 2. Although many image encryption methods have been proposed in the literature, they are not designed
primarily for addressing the technical requirements of 3D applications. In this chapter, the proposed encryption scheme is compared with AES, because AES is currently the main industrial encryption standard used in many multimedia applications. AES is a well-studied cipher and no practical attack has been found against it to date. In addition, AES is fast and on a Core 2 architecture, for example, runs around 12 cycles/byte for long streams. This speed is quite fast compared to other multimedia operations such as compression (For instance, Lempel-Zev and ZLIB Compressions). However, it is shown that AES cannot sufficiently address the confidentiality requirements of texture images, and therefore texture images encrypted by AES may leak crucial information about the protected 3D models. It is also shown that in comparison with AES, the proposed encryption scheme not only maintains the confidentiality of texture images but also maintains the security of protected 3D models from surface reconstruction attacks.

The remainder of this chapter is organised as follows. Section 5.2 describes the encryption and decryption procedures. In Section 5.3, the performance of the proposed cipher is evaluated. Section 5.4 evaluates the security of the cipher from the data level and the semantic level. Finally, Section 5.5 concludes that the proposed texture encryption method is secure, relatively lightweight and prevents the partial disclosure of the protected 3D surface geometry by the texture pattern.

5.2 Proposed Texture Encryption Scheme

In a true colour (24-bit) representation, 94.125% of the total information is stored in the upper nibbles (4 bit-planes) of the texture image. This suggests employing a strict strategy to encrypt the upper nibbles and a lenient scheme for the encryption of lower nibbles. This approach improves the encryption performance and reduces the memory usage. The proposed scheme encrypts the upper nibble-image using a fast stream cipher, that is, Salsa20/12 [166], and scrambles the bit stream of the lower nibble-image by a zigzag pattern permutation. This mechanism is consistent with the movement performed in the (Latin American) Salsa dance. We therefore call our encryption mechanism ‘Salsa Dance’.

To elaborate the steps of the encryption algorithm, denote by \( P \) the plain-image, \( N \) the nibble-image, and \( C \) the cipher-image. In a 24-bit true colour representation, each plain-image, nibble-image or cipher-image is represented by three \( M \times N \) matrices, namely, \( R \), \( G \), and \( B \) colour layers. In any colour layer of RGB, for any \( x \) (1 ≤ \( x \) ≤ \( M \)), and \( y \) (1 ≤ \( y \) ≤ \( N \)), let \( p(x, y) \), \( n(x, y) \) and \( c(x, y) \), be the entry value at the position \((x, y)\) of the plain-image, nibble-image and cipher-image, respectively. \( p(x, y) \) and \( c(x, y) \) ∈ {0, 1, ··· , 255}, and \( n(x, y) \) ∈ {0, 1, ··· , 15}.

The encryption procedure is described as follows, for one colour layer of a 24-bit texture image. The encryption procedure is the same for the other colour layers. Firstly, the plain-image is divided into two nibble-images, which are \( N_1 \) and \( N_2 \), by splitting every entry into
upper and lower nibbles. For any \( x \) \((1 \leq x \leq M)\) and \( y \) \((1 \leq y \leq N)\), \( n_1(x, y) \) and \( n_2(x, y) \) are defined as follows:

\[
\begin{align*}
    n_1(x, y) &= p(x, y) \mod 2^4, \quad (5.1) \\
    n_2(x, y) &= (p(x, y) - n_1(x, y)) \cdot 2^{-4}. \quad (5.2)
\end{align*}
\]

In the upper nibble-image encryption, the binary stream of the upper-nibble images with size \( M \times 4N \) is masked with a binary stream of the same size generated by the Salsa20/12 stream cipher. This procedure not only protects the coarse shape (major information) of the texture images from being leaked, but also prevents the disclosure of the 3D surface geometry. The stream cipher Salsa20/12 operates on 32-bit words, takes as input a 128-bit key \( k \) and a 64-bit nonce \( IV = (v_0, v_1) \), and produces a sequence of 512-bit keystream blocks. The \( i \)-th block is the output of the Salsa20/12 function, that takes as input the key, the nonce, and a 64-bit counter \( t = (t_0, t_1) \) corresponding to the integer \( i \). This function acts on the 4 \times 4 matrix of 32-bit words written as

\[
X = \begin{bmatrix}
    x_0 & x_1 & x_2 & x_3 \\
    x_4 & x_5 & x_6 & x_7 \\
    x_8 & x_9 & x_{10} & x_{11} \\
    x_{12} & x_{13} & x_{14} & x_{15}
\end{bmatrix}
= \begin{bmatrix}
    c_0 & k_0 & k_1 & k_2 \\
    k_3 & c_1 & v_0 & v_1 \\
    t_0 & t_1 & c_2 & k_0 \\
    k_1 & k_2 & k_3 & c_3
\end{bmatrix}.
\quad (5.3)
\]

The \( c_0, c_1, c_2 \) and \( c_3 \) are predefined constants, that is, \( c_0 = 0x61707865 \), \( c_1 = 0x3120646e \), \( c_2 = 0x79622d36 \), and \( c_3 = 0x6b206574 \). A keystream block \( Z \) is then defined as

\[
Z = (X + X^{12}) \mod 2^{32},
\quad (5.4)
\]

where \( X^{12} = \text{Round}^{12}(X) \) with the round function \( \text{Round} \) of Salsa20/12. The round function is based on the following nonlinear operation, namely the quarterround function, which transforms a vector \((x_0, x_1, x_2, x_3)\) to \((z_0, z_1, z_2, z_3)\) by sequentially computing

\[
\begin{align*}
    z_1 &= x_1 \oplus [(x_3 + x_0) \ll 7], \quad (5.5) \\
    z_2 &= x_2 \oplus [(x_0 + z_1) \ll 9], \quad (5.6) \\
    z_3 &= x_3 \oplus [(z_1 + z_2) \ll 13], \quad (5.7) \\
    z_0 &= x_0 \oplus [(z_2 + z_3) \ll 18], \quad (5.8)
\end{align*}
\]

where \( x \ll l \) denotes the \( l \)-bit left rotation of a word \( x \). In odd numbers of rounds, the nonlinear operation is applied to columns \((x_0, x_1, x_2, x_3), (x_5, x_9, x_{13}, x_1), (x_{10}, x_{14}, x_2, x_6), (x_{15}, x_3, x_7, x_{11})\). In even numbers of rounds, the nonlinear operation is applied to the rows \((x_0, x_1, x_2, x_3), (x_5, x_6, x_7, x_4), (x_{10}, x_{11}, x_8, x_9), (x_{15}, x_{12}, x_{13}, x_{14})\).
Chapter 5. Texture Encryption and Maintaining the Secrecy of Protected 3D Objects

In the lower nibble-image encryption, the lower nibble-image is first extended to a bit-plane image with size $M \times 4N$, which is constructed by expanding every column of the lower nibble-image into 4 bit-plane columns. The bit-plane image then undergoes a bit-level zigzag pattern permutation process $\text{Perm}(\cdot)$. Displacement of the bit locations not only annihilates the high correlation among the nibbles but also increases the security level of the encrypted texture images. This process is as follows:

Assume that the entries of the bit-plane image are scanned in a raster order and they are enumerated by positive integers. Let $\mathbf{R}$ denote the matrix of the entry (bit) locations, that is,

$$
\mathbf{R} = \begin{bmatrix}
0 & M & \cdots & 4MN - M \\
1 & M + 1 & \cdots & 4MN - M + 1 \\
\vdots & \vdots & \vdots & \vdots \\
M - 1 & 2M - 1 & \cdots & 4MN - 1
\end{bmatrix}.
$$

(5.9)

An additional binary sequence of length $\lceil \log_2 (4MN) \rceil$ with value $s$ is iterated by Salsa20/12. Then, $\text{mod} \ (s, MN)$ is used to select an entry in the bit-plane image, which determines the starting point for the zigzag-pattern permutation of the entries. To clarify further, Figure 5.1 shows a zigzag path for the scanning of entries in a bit-plane image with size $3 \times 4$. In Figure 5.1a, if $\text{mod} \ (s, 12) = 7$, then the entry scanning commences from the 7-th entry, and stops at the 9-th entry which is previous to the initial one, that is 7. During the scanning process, bits encountered in the path are arranged sequentially, column by column in the same matrix. On completion of the permutation process, not only is every bit dislocated (diffusion), but also nibble values are modified (confusion) within the bit-plane image. For $\text{mod} \ (s, 12) = 7$, the permutation result of the test bit-plane image is depicted in Figure 5.1b. Following the permutation process, the encrypted lower nibble-image with size $M \times N$ is reconstructed by combining every 4 consecutive columns of the scrambled bit-plane image. Finally, the cipher-image is constructed by the radix $2^4$ combination of the encrypted upper and lower nibble-images.

![Figure 5.1: (a) A zigzag path to scramble bits of a bit-plane image, (b) Permutation result of the bit-plane image for $\text{mod} \ (s, 12) = 7$.](image)

In summary, the whole encryption process $E(\cdot)$ is as follows:

$$
P = 2^4 \cdot N_2 + N_1,
$$

(5.10)
\[ C = E(P) = 2^4 \cdot E_2(N_2) + E_1(N_1), \]  
\( (5.11) \)

where
\[ E_2(N_2) = Salsa20/12(N_2), \]
\( (5.12) \)
\[ E_1(N_1) = Perm(N_1). \]
\( (5.13) \)

\( P, N_1, N_2, \) and \( C \) denote the plain-image, lower nibble-image, upper nibble-image and the cipher-image, respectively. In decryption, the cipher-image is divided into lower and upper nibble-images. The upper nibble-image is decrypted by the same key stream used in encryption, and the lower nibble-image is decrypted by the inverse permutation procedure. Note that in 24-bit texture images there is a strong correlation among different colour layers of the image. Therefore, encryption of different colour layers using the same key may reveal the underlying pattern. To address this issue, Salsa20/12 has a 64-bit nonce which changes after each colour layer encryption. This ensures that whenever the same message is encrypted twice, the ciphertext is always different. If the same nonce and key are used to encrypt two different plaintexts, then the keystream can be cancelled out by masking the two different ciphertexts together.

### 5.3 Performance Analysis

To evaluate the performance of the proposed cipher, we implemented

1. The full encryption by 128-bit AES,
2. Selective encryption of the 4 most significant bit-planes using 128-bit AES, and
3. Salsa Dance,

on a machine with Intel Core 2 2.4 GHz processor and 4 GB of installed memory. We tested the encryption performance using 500 sample texture images from CGTextures [168]. Figure 5.2 shows one test texture image with its corresponding encryption results. It is observed that Salsa Dance significantly dissipates the correlation among the entries of the texture image while the full and selective encryption using AES cannot annihilate the coarse pattern of the texture image.

In the proposed encryption method, 24-bit texture images with size \( MN \) (that is, \( 24 \times MN \) bits in total) are encrypted by a pseudorandom binary sequence with size \( 12MN + 3\lceil \log_2 (4MN) \rceil \). In other words, the proposed cipher encrypts the input data by generating a pseudorandom sequence with the size of almost 50% of the data. This means that compared to conventional full encryption methods, the proposed method reduces the computational cost to approximately half. This reduction in the computational cost can therefore save computational power, storage space, processing time, and transmission bandwidth; and, therefore, it would allow more processes to be executed in parallel.
Chapter 5. Texture Encryption and Maintaining the Secrecy of Protected 3D Objects

Figure 5.2: Encryption results of a sample texture image: (a) original image, (b) encrypted image using full AES, (c) encrypted image using selective AES, (d) encrypted image using Salsa Dance.

Table 5.1: Comparison of the relative CPU time

<table>
<thead>
<tr>
<th>Encryption schemes</th>
<th>Relative CPU time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Selective AES</td>
<td>2.47</td>
</tr>
<tr>
<td>Full AES</td>
<td>4.95</td>
</tr>
<tr>
<td>Proposed (Salsa Dance)</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Compared to the 10 rounds 128-bit AES, Salsa20/12 is considerably faster. On a Core 2 architecture, for example, Salsa20/12 runs at 2.54 cycles/byte for long streams, while the fastest speed reported for 128-AES is 12.59 cycles/byte [169]. This implies that Salsa20/12 is almost 5 times faster than the 10 rounds 128-bit AES. Therefore, Salsa20 provides a much better speed-security profile than AES. To evaluate the encryption speed of the proposed cipher, numerous encryption timing tests were performed. In addition, to have an accurate benchmark result, each timing test was executed 10 times and the average time was recorded. The results of timing tests demonstrated that the 4 bit-planes selective encryption methods have speed overheads of 247% on average compared to Salsa Dance. Also, the full AES schemes have 495% speed overheads compared to Salsa Dance. Table 5.1 compares the execution time of the encryption methods. Hence, the experimental results indicate that Salsa Dance has a better encryption performance than the full and selective encryption using AES.

5.4 Security Analysis

From the data level point of view, the security of encryption, including the upper nibble method and the lower nibble method, relies on the security of the encryption primitive, that is, Salsa20/12. To the best of the authors’ knowledge, the best cryptanalysis breaks 8 out of 20 rounds of Salsa20 to recover the 256-bit secret key in $2^{251}$ operations, using $2^{31}$ keystream pairs [170]. Also, it is conjectured that Salsa20 and AES reach security with about the same number of rounds [166]. This means that the upper nibble encryption method offers a high confidentiality level. In addition, the lower nibble encryption method, that is, the permutation procedure, is secure as well, because the pseudorandom key stream controlling the permutation is generated by Salsa20/12.
The generated key stream is different even for the same colour layer encrypted at different sessions. This makes the permutation scheme robust to known/chosen plaintext attacks. Hence, the only attack model applicable to the permutation method is the ciphertext-only attack [171], in which the attacker can only access the lower nibble-image of the cipher-image and attempts to recover the lower nibble-image of the original image by trying all possible permutations (4MN possible arrangements in each colour layer). This attack becomes cumbersome and even impractical by increasing the input size MN (This increases the data complexity of the attack).

However, from the semantic level point of view, encrypted texture images may contain redundant information which may be employed to not only retrieve the original texture images but also to reconstruct 3D objects. To evaluate the security of encryption to redundancy based attacks, several measurements were performed, including a correlation analysis, a key sensitivity analysis, and an edge detection analysis. Each of these measurements is described in detail in the following subsections.

### 5.4.1 Correlation Analysis

In the texture images, each pixel is highly correlated with its adjacent pixels. Therefore, the adversary may study the correlation among the pixels to determine a meaningful pattern inside the encrypted texture image. An ideal encryption algorithm should completely dissipate such relationship and produce cipher-images with no correlation in the adjacent pixels. A correlation of a pixel with its neighbouring pixel is then given by a 2-tuple \((x_i, y_i)\) where \(y_i\) is the adjacent pixel of \(x_i\). The following equation is used to study the correlation between two adjacent pixels in horizontal, vertical and diagonal orientations.

\[
\text{corr}(x,y) = \frac{1}{n-1} \sum_{i=0}^{n} \left( \frac{x_i - \bar{x}}{\sigma_x} \right) \left( \frac{y_i - \bar{y}}{\sigma_y} \right),
\]

where \(x\) and \(y\) are intensity values of two neighbouring pixels in the image, \(n\) represents the total number of 2-tuples \((x_i, y_i)\), and \(\sigma_x\) and \(\sigma_y\) represent the local standard deviation, respectively.

To test the impact of encryption by Salsa Dance on the correlation among the adjacent pixels, we performed several correlation tests. Figure 5.3 shows the correlation distribution of two adjacent pixels in the plain-image shown in Figure 5.2.a and its corresponding cipher-image. It is observed that neighbouring pixels in the plain-image are highly correlated, while the neighbouring pixels in the encrypted image are almost uncorrelated. Table 5.2 shows the results for correlation coefficients of the ciphers under study. The numerical results indicate that the correlation coefficients of plain-images are far apart from cipher-images. Also, results show that the selective/full AES and Salsa Dance efficiently dissipate the correlation among pixels within each colour layer. Furthermore, we computed the 2D correlation coefficients between every two colour layers of the encrypted images. Table 5.3 shows the correlation coefficients between different colour layers of the cipher-images produced by the AES encryption and Salsa dance.
It is observed that Salsa Dance can reduce the strong correlation between the colour layers much better than the encryption using 128-bit AES. Hence, the results of the correlation analysis indicate that compared to the full and selective encryption using 128-bit AES, Salsa Dance has a better encryption performance and is more robust to redundancy based attacks.

5.4.2 Key Sensitivity Analysis

It is possible for an adversary to induce modifications in the secret key via tampering or fault injection [172]. This helps the adversary to observe the redundancy under different encryption keys and deduce a relationship between the used keys. To resist such kinds of analyses, a texture image encryption scheme should be sensitive to changes to the secret key. In other words, a change in a single bit of the secret key should produce a completely different cipher-image. The more the visual data is sensitive toward the secret key, the higher would be the amount of data randomness. To test the key sensitivity of the proposed algorithm, a number of texture images were encrypted using the selective/full AES and Salsa Dance with an original secret key ($K = 0$, $IV = 0$) and a slightly modified secret key ($K' = 1$, $IV = 0$). Numerical results show that the proposed technique is highly sensitive toward the small alterations of the secret key, that is, a different cipher-image is produced when the secret key is slightly changed. For comparison purposes, we used the PSNR measure. The higher the PSNR, the closer the images are. Table 5.4 provides the PSNR values of the encrypted images for the test image shown in Figure 5.2.a. It is observed that given the test image shown in Figure 5.2.a, encryption by slightly different secret keys creates different cipher-images by the selective/full AES and Salsa Dance. However, Table 5.4 shows that compared to the encryption by selective/full AES, Salsa Dance produces more dissimilar cipher-images with only 1-bit of change in the secret key. This indicates the high sensitivity of the proposed method to changes of the key, which makes the analysis of Salsa Dance even harder for the adversary in respect to finding any relationship between the used keys.

5.4.3 Edge Detection Analysis

From the semantic point of view, the coarse pattern of the visual data (that is, the shape information) carries more information than the details. Disclosure of the shape information not only may help a competent adversary in retrieving the texture image but also may facilitate the reconstruction of underlying 3D objects. Therefore, the adversary would attempt to identify and locate the boundaries of the protected object within the encrypted texture images. The object boundaries, as well as sharp variations in surface structure, are typically manifested by sharp changes in pixel intensities. However, the randomness of encrypted images makes the edge detection hard. To this end, the adversary may use nonlinear operations, such as median filtering,
Chapter 5. *Texture Encryption and Maintaining the Secrecy of Protected 3D Objects*

**Figure 5.3:** Correlation analysis and distribution of two adjacent pixels in the plain-image and cipher-image.
## Table 5.2: Correlation coefficients of two adjacent pixels in plain-image and cipher-image

<table>
<thead>
<tr>
<th>File name</th>
<th>Size</th>
<th>Channel</th>
<th>Plain-image</th>
<th>Cipher-image</th>
<th>Proposed (Salsa Dance)</th>
<th>Full AES</th>
<th>Selective AES</th>
<th>Plain-image</th>
<th>Cipher-image</th>
<th>Proposed (Salsa Dance)</th>
<th>Full AES</th>
<th>Selective AES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fruit</td>
<td>00562800 × 2656</td>
<td>R</td>
<td>0.9879</td>
<td>0.9882</td>
<td>0.9741</td>
<td>0.0542</td>
<td>0.0279</td>
<td>0.0541</td>
<td>0.0138</td>
<td>0.0179</td>
<td>0.0078</td>
<td>0.0164</td>
</tr>
<tr>
<td>Buildings</td>
<td>00045000 × 3464</td>
<td>R</td>
<td>0.9863</td>
<td>0.9874</td>
<td>0.9777</td>
<td>0.0133</td>
<td>0.1278</td>
<td>0.0143</td>
<td>0.0238</td>
<td>0.0041</td>
<td>0.0041</td>
<td>0.0287</td>
</tr>
<tr>
<td>Buildings</td>
<td>000881600 × 1024</td>
<td>R</td>
<td>0.9359</td>
<td>0.9325</td>
<td>0.8701</td>
<td>0.0953</td>
<td>0.1019</td>
<td>0.0538</td>
<td>0.0286</td>
<td>0.0360</td>
<td>0.0527</td>
<td>0.0297</td>
</tr>
<tr>
<td>Bones</td>
<td>0009936000 × 1024</td>
<td>R</td>
<td>0.9793</td>
<td>0.9706</td>
<td>0.9521</td>
<td>0.0129</td>
<td>0.0159</td>
<td>0.0061</td>
<td>0.0159</td>
<td>0.0445</td>
<td>0.0409</td>
<td>0.0118</td>
</tr>
<tr>
<td>Bones</td>
<td>00081192000 × 1600</td>
<td>R</td>
<td>0.9753</td>
<td>0.9666</td>
<td>0.9435</td>
<td>0.0107</td>
<td>0.0211</td>
<td>0.0045</td>
<td>0.0075</td>
<td>0.0071</td>
<td>0.0121</td>
<td>0.0334</td>
</tr>
<tr>
<td>Gobos</td>
<td>0125300000 × 2000</td>
<td>R</td>
<td>0.8258</td>
<td>0.9066</td>
<td>0.8298</td>
<td>0.0688</td>
<td>0.3088</td>
<td>0.0624</td>
<td>0.0586</td>
<td>0.0315</td>
<td>0.0445</td>
<td>0.0223</td>
</tr>
<tr>
<td>Gobos</td>
<td>0122160000 × 1184</td>
<td>R</td>
<td>0.8638</td>
<td>0.9034</td>
<td>0.8177</td>
<td>0.0700</td>
<td>0.2376</td>
<td>0.0457</td>
<td>0.0189</td>
<td>0.0192</td>
<td>0.0066</td>
<td>0.0120</td>
</tr>
<tr>
<td>BookSide</td>
<td>002712320000 × 3000</td>
<td>R</td>
<td>0.9547</td>
<td>0.9992</td>
<td>0.9624</td>
<td>0.0504</td>
<td>0.4226</td>
<td>0.0038</td>
<td>0.0075</td>
<td>0.0552</td>
<td>0.0015</td>
<td>0.0049</td>
</tr>
<tr>
<td>BookSide</td>
<td>0031160000 × 648</td>
<td>R</td>
<td>0.9703</td>
<td>0.9981</td>
<td>0.9547</td>
<td>0.0330</td>
<td>0.4261</td>
<td>0.0185</td>
<td>0.0042</td>
<td>0.0563</td>
<td>0.0481</td>
<td>0.0723</td>
</tr>
<tr>
<td>Windows Shutters</td>
<td>011451840000 × 3456</td>
<td>R</td>
<td>0.8373</td>
<td>0.9898</td>
<td>0.8334</td>
<td>0.0383</td>
<td>0.5307</td>
<td>0.0013</td>
<td>0.0229</td>
<td>0.1393</td>
<td>0.0238</td>
<td>0.0536</td>
</tr>
<tr>
<td>Windows Shutters</td>
<td>0096160000 × 1048</td>
<td>R</td>
<td>0.8730</td>
<td>0.9866</td>
<td>0.8526</td>
<td>0.0087</td>
<td>0.5528</td>
<td>0.0158</td>
<td>0.0029</td>
<td>0.1292</td>
<td>0.0138</td>
<td>0.0249</td>
</tr>
</tbody>
</table>
Table 5.3: 2D correlation coefficients between the RGB colour layers of the cipher-images

<table>
<thead>
<tr>
<th>Encryption schemes</th>
<th>Selective AES</th>
<th>Full AES</th>
<th>Proposed (Salsa Dance)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between R and G</td>
<td>0.3639</td>
<td>0.3237</td>
<td>0.0016</td>
</tr>
<tr>
<td>Between R and B</td>
<td>0.3484</td>
<td>0.3041</td>
<td>0.0050</td>
</tr>
<tr>
<td>Between G and B</td>
<td>0.3834</td>
<td>0.3041</td>
<td>0.0018</td>
</tr>
</tbody>
</table>

Table 5.4: Comparison of the PSNR values

<table>
<thead>
<tr>
<th>Encryption schemes</th>
<th>Selective AES</th>
<th>Full AES</th>
<th>Proposed (Salsa Dance)</th>
</tr>
</thead>
<tbody>
<tr>
<td>between the original and encrypted image with original key</td>
<td>6.0403dB</td>
<td>6.0345dB</td>
<td>6.0839dB</td>
</tr>
<tr>
<td>between the original and encrypted image with 1-bit different key</td>
<td>6.2709dB</td>
<td>6.1846dB</td>
<td>6.0821dB</td>
</tr>
<tr>
<td>between the encrypted images using the original and modified keys</td>
<td>8.5122dB</td>
<td>8.3964dB</td>
<td>7.7680dB</td>
</tr>
</tbody>
</table>

to reduce the noise while maintaining the edges. He/she may then employ gradient and Laplacian operators for the edge-detection. This information is essential for the correct reconstruction of 3D surfaces [173]. To evaluate the resistance of the proposed texture encryption scheme to this kind of analysis, we examined the cipher-images using different edge detection methods [174], including the Canny method. Figure 5.4 shows the results of the edge detection analysis on the cipher-images of Figure 5.2 using the Canny method. It is observed that Salsa Dance discloses no information about the shape and boundaries of the underlying 3D object, while the full and selective encryption using AES cannot resist the edge detection analysis.

![Figure 5.4: Results of edge detection by median and Canny filtering for the (a) encrypted image using full AES, (b) encrypted image using selective AES, and (c) encrypted image using Salsa Dance.](image)
5.5 Conclusion

To overcome the limitations of the current techniques in addressing the confidentiality requirement of texture images, this chapter proposes a technical solution that meets the constraints imposed by the structure of texture images, such as large data volume, and the application requirements, such as real-time performance, complexity, and the security level. The proposed cipher encrypts texture images by bit masking and a permutation procedure using the Salsa20/12 stream cipher. Compared to the full/selective encryption using 128-bit AES, the proposed cipher is relatively lightweight and provides a better encryption performance. Salsa Dance also considerably dissipates the correlation among the entries of the texture image. This annihilates the coarse pattern of the plain-image and prohibits the data leakage from texture images. The key sensitivity analysis showed that even a single bit change in the secret key will result in an entirely different cipher-image. Thus, the original texture image cannot be recovered even though there is a slight difference between the encryption and decryption keys. Furthermore, Salsa Dance conceals the shape and boundaries of the underlying 3D object, while the full and selective encryption using AES is not secure from the edge detection analysis. Therefore, texture encryption by Salsa Dance not only maintains the confidentiality of texture images but also maintains the security of protected 3D models from surface reconstruction attacks using the data provided by the texture images.

In the next chapter, we analyse the security of permutation-only encryption primitives, and propose a practical chosen-plaintext attack that completely breaks permutation-only image encryption schemes. The strategy of this new attack is further expanded to break more complex ciphers which are constructed by the combination of a permutation primitive with a substitution primitive.
6

Cryptanalysis of Permutation-Only Image Encryption Schemes

6.1 Introduction

The fast growing demand for digital multimedia applications has opened up a number of challenges regarding the confidentiality of images and videos in many multimedia-based services, such as Pay-TV, remote video conferencing, and medical imaging. Reliable storage and secure transmission of visual content is a legitimate concern of IP owners. Thus, there is a strong need to protect images and videos against unauthorised use or other security violations. Encryption is a solution to maintain confidentiality. Multimedia encryption obfuscates the image/video data-stream to ensure secure transmission of image/video data between two parties over a public channel. Given the fact that raw video data is constructed by a sequence of still images (frames), image encryption techniques can be applied to still images or single frames in a video.

The problem of image encryption is beyond the application of established and well-known encryption algorithms. This is primarily due to the constraints imposed by the data structure and the application requirements, such as format compliance [175], real-time performance [176], complexity [177], compression efficiency [178], perceptibility [179] and the security level [180]. To address these concerns, significant attempts have been made to develop robust encryption schemes for the image data [148], [181], [182].
Due to the grid structure of digital images, image encryption methods utilise three different types of operations: position permutation, value transformation, and the combination form. Among different operations, permutation (transposition) is a commonly used primitive in many image encryption schemes. This is mainly due to the easy implementation and applicability of permutation in both spatial and frequency domains. In addition, by combining permutation with other simple value transformation operations, such as XOR, a highly secure multimedia encryption scheme can be achieved. In all the well-known permutation-only ciphers, image entries (or bit-planes) are permuted by a mapping matrix which is built by a pseudo-random number generator. From the design point of view, permutation dissipates the statistical structure of the plaintext into long range statistics and it is suitable for fast processing requirements of massive digital multimedia data [183], [184].

Despite the advantages of permutation, it has a number of inherent limitations. Permutation-only ciphers disclose some essential characteristics of the plaintext, such as the frequency distribution of symbols in the plaintext. Also, when the size of plaintext is small, that is, the number of possible arrangements for the plaintext elements is less than the key space, the number of effective keys can be reduced, and hence, the permutation mapping can be disclosed. Moreover, permutation-only encryption/decryption are not simple sequential operations that can be done dynamically. In general, permutation may need a buffer with a size comparable to that of the plaintext. Therefore, due to the limitations above, permutation-only ciphers are nowadays only used in applications where substitution is technically infeasible and/or only a moderate level of protection is required. Considering typical examples of permutation-only image ciphers, in [185], [186], and [187] image entries are dislocated using pseudo-random permutations; in [188] and [189] permutation operations are performed on the bit-planes of the image entries; and in [190] and [191] permutation operations are performed on DCT/wavelet coefficients.

The security of permutation-only image encryption schemes has been studied for a long time, and it has been shown that most of such schemes are insecure against ciphertext-only attacks and/or known/chosen-plaintext attacks, which is due to the high information redundancy in the multimedia data and some specific weaknesses in the encryption algorithms [192], [193], [194]. Despite the extensive cryptanalysis of permutation-only multimedia ciphers, in recent years, many permutation-only ciphers have been proposed for the protection of multimedia data, including digital images [186], [188], [189] and video [187], [190], [191]. This is mainly because the above-mentioned cryptanalytic methods can only be applied to specific encryption methods and cannot be generalized to a wider class of permutation-only multimedia ciphers [195], [196], [197], [198]. In addition, even the best known methods of known/chosen-plaintext attacks ([108] and [199]) cannot ensure the complete retrieval of the correct plaintext content, and hence, it is still ambiguous as to whether the security of permutation-only image ciphers can be effectively improved by designing new methods to generate better pseudo-random permutations.
This chapter presents a cryptanalysis which breaks most (if not all) permutation-only multimedia ciphers. In fact, it is shown that all permutation-only image ciphers are completely broken by chosen-plaintext attacks and no better pseudo-random permutation mapping can be realised to offer a higher level of security against chosen-plaintext attacks. For a successful attack, we derived a tight lower bound for the required number $n$ of chosen plain-images, that is, $n = \lceil \log_L (MN) \rceil$, comparing to the currently known results $O \left( \lceil \log_L (MN) \rceil \right)$ [108], [199], where $MN$ is the size of the image and $L - 1$ is the maximum colour intensity, that is, a colour intensity is specified by $l \ (0 \leq l \leq L - 1)$. The computational complexity of the proposed attack is $O (n \cdot MN)$. To verify the feasibility of the proposed attack, experiments were performed on the recently proposed permutation-only image ciphers by Rahman et al. [187] and Fu et al. [188]. Our experimental results support the theoretical results that pseudo-random permutations alone cannot provide sufficient security against chosen-plaintext attacks. Compared to the state of the art cryptanalytic methods of [108] and [199], which partially (quantitatively) determine the permutation mapping, our chosen-plaintext attack gives a precise procedure for the careful construction of the required chosen plain-images, and therefore, completely discloses the correct permutation mapping with less data and computational complexity.

The rest of this chapter is organised as follows. Section 6.2 reviews the related work in the cryptanalysis of permutation-only image ciphers. In section 6.3, the procedure of the chosen-plaintext attack is described. Section 6.4 overviews two typical permutation-only image ciphers (case studies) proposed by Rahman et al. [187] and Fu et al. [188]. Experimental results are shown in Section 6.5 to support the theoretical cryptanalysis. Section 6.6 discusses the advantages of the proposed chosen-plaintext attack in comparison to the state of the art cryptanalyses. Finally, the last section concludes the chapter.

6.2 Previous Works in Cryptanalysis of Permutation-Only Image Ciphers

The security of permutation-only image ciphers has been extensively studied. These cryptanalytic studies are briefly described as follows. In [195], Matias and Shamir analysed the security of early permutation-only image encryption schemes used in analogue broadcasting systems. The prominent feature of such ciphers were that they utilised fewer numbers of permutations with shorter domains, with the intention of keeping the bandwidth increase of the encryption process as low as possible. This made the early permutation-only image encryption schemes more vulnerable to correlation attacks, implying that the high correlation properties remaining in the permuted images could be employed to restore the image. To address the correlation issues, Matias and Shamir proposed a permutation-only scheme which scanned pixels in a highly irregular scanning pattern using a pseudo-random space filling curve. Bertilsson et al. [196] then showed that Matias and Shamir’s permutation method is vulnerable to a ciphertext-only...
attack. They showed that the pixel data could be reordered according to a space-filling curve, and hence, the plain-image could be partially recovered by exploiting the correlation between subsequent frames.

Later, Kuhn [197] presented a more advanced approach to break the video signal scramblers commercially employed within pay-TV conditional access encryption systems [200], such as EuroCrypt, VideoCrypt and Nagravision, using ciphertext-only attacks. Kuhn showed that the long portion of the permuted lines/segments makes the correlation attacks on the scrambling algorithm feasible by comparing and matching lines/segment portions. Li et al. [198] then extended Kuhn’s work by analysing the permutation domain of particular image encryption schemes with longer permutation domains, such as the row-column permutation-only encryption scheme of [185]. Despite the efforts made to improve the performance of previous ciphertext-only attacks, these attacks are only applicable to schemes whose permutation domains are considerably smaller than the size of input images. Indeed, increasing the permutation domain makes the correlation analysis, and hence the ciphertext-only attacks, computationally cumbersome.

To reduce the complexity of the exhaustive key search (a ciphertext-only attack), Li et al. [108] provided a general cryptanalysis (a known-plaintext attack and a chosen-plaintext attack) based on the quantitative relation between the breaking performance and the number of required known/chosen plaintexts. They showed that the number $n$ of required known/chosen plain-images to perform a successful known/chosen-plaintext attack on a permutation-only cipher is $O\left(\lceil \log_L (MN) \rceil \right)$, where $MN$ is the size of the image and $L$ is the number of colour intensities. They also detailed a procedure for the implementation of their attack which has $O\left(n (MN)^2\right)$ complexity, where $n$ is the number of known/chosen plain-images. Further, Li and Lo [199] improved the implementation performance of Li et al.’s cryptanalysis by reducing its computational complexity to $O\left(n (MN)\right)$. As explained in [199], the improvement in computational complexity is obtained by employing a multi-branch tree instead of the complex intersection operations in Li et al.’s attack. Despite the good recovering performance of the Li et al.’s cryptanalysis, it is not complete and cannot precisely identify the correct elements of the input plain-images with regard to chosen-plaintext attacks. This is mainly because Li et al.’s cryptanalysis is under the assumption of a uniform distribution of all entries in the plain-image. The distribution of colour intensities in most natural images is not uniform. More importantly, as explained in [108], Li et al.’s cryptanalysis can only determine a portion of the correct elements, that is, almost half of the elements, and predicts the other elements either by using image processing techniques or by inputting additional plain-images. Indeed, finding the exact value of unknown elements of an image by its partially known elements is hard.
6.3 Proposed Chosen-Plaintext Attack

Before we elaborate the proposed chosen-plaintext attack, the following definitions are given to describe a permutation-only image cipher.

**Definition 6.1.** Let \( S = \{ s \mid s = 0, 1, \ldots, MN - 1 \} \) denote the set of entry locations for an image with size \( M \times N \).

**Definition 6.2.** Assume that locations of image entries are scanned in a raster order and they are enumerated by non-negative integers, which are chosen from the set of entry locations. Let \( R \) denote the matrix of entry locations, that is,

\[
R = \begin{bmatrix}
0 & 1 & \cdots & N - 1 \\
N & N + 1 & \cdots & 2N - 1 \\
\vdots & \vdots & \ddots & \vdots \\
(M - 1)N & (M - 1)N + 1 & \cdots & MN - 1
\end{bmatrix}.
\]  
(6.1)

**Definition 6.3.** Let \( P \) and \( C \) denote the plain-image and cipher-image, respectively. Note that each plain-image or cipher-image is represented by an \( M \times N \) matrix, where the entry of such a matrix at position \( s \) corresponds to colour intensity. For any \( s \) \((0 \leq s \leq MN - 1)\), let \( p(s) \) and \( c(s) \) be the colour intensities at the position \( s \) of the plain-image and cipher-image, respectively.

**Definition 6.4.** Let \( X \) be a finite set. Permutation \( \Pi_k : X \to X \) is a bijection which maps the elements of \( X \) to itself. Each secret key \( k \in K \) assigns a different permutation.

**Definition 6.5.** A permutation-only image cipher \( \rho \) is defined by a permutation which, given a secret key \( k \), maps any entry location \( s \) \((0 \leq s \leq MN - 1)\) of a plain-image to its corresponding location \( \rho_k(s) \) in the cipher-image, where \( \rho_k \) is a permutation determined by \( k \).

The permutation-only image cipher is pseudo-random if it permutes the location of plain-image entries, with an approximate uniform probability, from the set of all possible \((\#S)\)! arrangements.

Let us now explain the procedure of the proposed chosen-plaintext attack. Deducing the permutation mapping \( \rho_k \) is equivalent to finding the secret key \( k \). Hence, the problem of breaking the cipher is defined as an attempt to deduce the permutation mapping without any prior knowledge of the key. Consider the adversary as an oracle machine which has access to the encryption and decryption functions, that is, \( \rho_k \) and \( \rho_k^{-1} \). The adversary asks \( n \) number of \( \rho_k \) or \( \rho_k^{-1} \) queries to obtain a set of \( n \) plain-image and cipher-image pairs, that is, \( \partial = \{(P_i, C_i) \mid i = 1, 2, \ldots, n\} \).

**Proposition 6.1.** For any \( i \) \((1 \leq i \leq n)\) and \( j \) \((1 \leq j \leq n)\), if either \( P_i = P_j \) or \( C_i = C_j \), then \( i = j \) and pairs \((P_i, C_i)\) and \((P_j, C_j)\) are identical.

**Proof.** This proposition is an obvious result, because the cipher is defined by a bijective permutation. ■
Definition 6.6. Given \( n \) pairs of plain-images and cipher-images, namely, \((P_1, C_1), (P_2, C_2), \ldots, (P_n, C_n)\), for any pair number \( r \) (\( 1 \leq r \leq n \)), source location \( s \) (\( 0 \leq s \leq MN - 1 \)), target location \( t \) (\( 0 \leq t \leq MN - 1 \)), and colour intensity \( l \) (\( 0 \leq l \leq L - 1 \)), where \( MN \) is the size of the image and \( L - 1 \) is the maximum colour intensity, the equivalent set \( J_r(s) \) is defined as a set of target locations in the \( r \)-th cipher-image, whose values are equal to the colour intensity \( l \) of the \( s \)-th location in the \( r \)-th plain-image, that is,

\[
J_r(s) = \{ t \mid c_r(t) = p_r(s), (0 \leq t \leq MN - 1) \}.
\] (6.2)

Obviously, by definition, the following condition holds for the equivalent sets:

\[
\bigcup_{s=0}^{MN-1} J_r(s) = \{ t \mid t = 0, 1, \ldots, MN - 1 \}.
\] (6.3)

For any \( r \) (\( 1 \leq r \leq n \)), each pair of plain-images and cipher-images, that is, \((P_r, C_r)\), involves two matrices with values assigned to entries. Consider the set \( S \) of entry locations in the plain-image. As explained in the beginning of this section, the permutation mapping \( \rho \) (see Definition 5) maps the source locations in the plain-image to the target locations in the cipher-image. To uniquely determine the permutation mapping, it is sufficient to study the arrangement of distinct entries in the pair of plain-images and cipher-images. In the case that all entries are assigned distinct values, the permutation is uniquely determined by a single pair. However, the set of colour intensities, that is, \( \{0, 1, \ldots, L - 1\} \), is finite and the images under study may have more than \( L \) entries. Therefore, for any \( r \) (\( 1 \leq r \leq n \)) and \( s \) (\( 0 \leq s \leq MN - 1 \)), by the pigeonhole principle the cardinality of some equivalent sets \( \#J_r(s) \) may not equal 1, and it is thus difficult to deduce a unique permutation mapping by knowing only one pair of plain-images and cipher-images. Hence, we need to have enough pairs of plain and cipher-images to determine the target location where each source location is mapped into. Therefore, the interest lies in using a collection of pairs, all of which have repeated values, to uniquely determine the underlying permutation. Clearly, the mapping of location \( s \) is uniquely determined if for any \( s \) (\( 0 \leq s \leq MN - 1 \)) and \( r \) (\( 1 \leq r \leq n \)), the equivalent sets \( J_r(s) \) intersect in a singleton, that is, \( \bigcap_{r=1}^{n} J_r(s) = \{ \rho(s) \} \), and hence it is sufficient to determine the permutation \( \rho \) if this is true for all \( s \). Two further questions then appear:

- Is this condition sufficient to determine unique \( \rho \)?
- With what accuracy and computational cost can the mapping \( \rho \) be determined from sufficient pairs?

To answer these questions, we need to find a relationship among the number of plain-image/cipher-image pairs \( n \), the number of locations \( MN \) and the number of assigned values in the locations \( L \). To perform a successful chosen-plaintext attack, it is necessary to find a
lower bound on the number of required pairs. However, it is possible for two given pairs to be related by a permutation on the colour intensities, such that both pairs give the same information regarding possible plain-image and cipher-image locations. Thus, a useful bound on the number of required pairs will entail some restriction that avoids this possible redundancy.

A best case in connection with lower bounds on pairs can be sharply stated as follows:

**Lemma 6.1.** Given $L$ colour intensities and $MN$ locations, for any permutation $\rho$, which is applied to get the respective cipher-images, there exist $n \geq \lceil \log_L(MN) \rceil$, such that $\rho$ is uniquely determined by making use of $n$ pairs of plain-images and cipher-images.

**Proof.** Consider $\lceil \log_L(MN) \rceil$ plain-images constructed by the $\lceil \log_L(MN) \rceil$ digit expansions in radix $L$ for $s = 0, 1, \ldots, MN - 1$ in respective locations. Taken the positional digits sequentially, these values uniquely label each of the $MN$ locations, and therefore $\rho$ is uniquely determined by finding the target locations which exactly match the source labelling. For instance, if $M = N = L = 2$, then 2 plain-images can be constructed by 2 digit expansions in radix 2 for $s = 0, 1, 2, 3$, that is, $s' = 00, 01, 10, 11$. The construction procedure of the chosen plain-image/cipher-image pairs is depicted in Figure 6.1.

If fewer pairs are used, that is, $n < \lceil \log_L(MN) \rceil$, then by counting the possible sequences of $L$ values for each location, it is easy to verify that there would be less numbers than $MN$ available locations. Thus, by the pigeonhole principle at least two locations would get the same source values in all pairs. It follows for any permutation $\rho$ that we would be unable to distinguish between the mapped target locations.

We can now prove the following result.

**Theorem 6.1.** The number of required chosen plain-images $n$ to perform a successful chosen-plaintext attack on a permutation-only image encryption algorithm is $n = \lceil \log_L(MN) \rceil$.

**Proof.** This theorem is an obvious result of Lemma 6.1. Theoretically, the permutation mapping can be easily deduced using an input matrix of size $MN$ whose entries are sequentially labelled.
with distinct values \(0, 1, \ldots, MN\). However, this is not practical because the encryption/decryption machine is only defined for entries of at most \(L - 1\), which is usually less than the number of entries. Therefore, to make the attack feasible, the entries are firstly expanded by \([\log_L (MN)]\) digits with radix \(L\). This matrix is then separated into \([\log_L (MN)]\) numbers of plain-images based on the digit positions in radix \(L\). Once permutation \(\rho\) is applied to the plain-images, it produces \([\log_L (MN)]\) cipher-images with entries in radix \(L\). A combination of cipher-images using the positional digits reveals the mapped locations of the original locations.

To illustrate the attack procedure, consider a \(5 \times 5\) matrix case.

1. If \(L = 1\), no further progress can be made toward determining the permutation, since the only plain-image/cipher-image pair has all entries assigned equal values.

2. If \(L = 2\), then the permutation can be determined by \([\log_2 (25)] = 5\) pairs of plain-images/cipher-images. One way to see this is to construct an input matrix \(P_1\) with 5-bit binary expansions for the 25 locations \(s = 0, 1, \ldots, 24\):

\[
P_1 = \begin{bmatrix}
00000 & 00001 & 00010 & 00011 & 00100 \\
00101 & 00110 & 00111 & 01000 & 01001 \\
01010 & 01011 & 01100 & 01101 & 01110 \\
01111 & 10000 & 10001 & 10010 & 10011 \\
10100 & 10101 & 10110 & 10111 & 11000
\end{bmatrix}.
\] (6.4)

Splitting this matrix into five binary source matrices based on bit positions, and application of the permutation \(\rho\) to these, produces five binary target matrices. When these matrices are recombined using positional bits, the mapped locations of the original locations \(s = 0, 1, \ldots, 24\) will be revealed.

3. If \(L = 3\), then a similar treatment requires only \([\log_3 (25)] = 3\) plain-image/cipher-image pairs. The original locations \(s = 0, 1, \ldots, 24\), can be expanded to 3 digits in ternary representation. Hence,

\[
P_2 = \begin{bmatrix}
000 & 001 & 002 & 010 & 011 \\
012 & 020 & 021 & 022 & 100 \\
101 & 102 & 110 & 111 & 112 \\
120 & 121 & 122 & 200 & 201 \\
202 & 210 & 211 & 212 & 220
\end{bmatrix}.
\] (6.5)

Then, plain-images whose entries are 0, 1 and 2 are generated by splitting this matrix into three. Cipher-images are then generated by applying the permutation to all three plain-images. Recombining target matrices as radix 3 values gives the permuted locations of \(s = 0, 1, \ldots, 24\), as required to determine the permutation.
4. Until one gets $L \leq 24$, more than one pair is necessary to deduce the permutation, as per the pigeonhole principle, some value has to be used more than once in a pair.

Next, we discuss whether it is possible to maximize the attack performance by choosing fewer than $\lceil \log_L (MN) \rceil$ pairs. This can only happen when the available pairs are well chosen. However, finding the exact minimum number of pairs to deduce permutation mapping is equivalent to the classic problem of the Test Cover \[201\], where a pair of a set of elements and a collection of subsets of the elements, named tests, are given. This problem is to determine the minimum sized subset of a collection of sets such that for every pair there is a test in the selection that contains exactly one of the two elements. It has been proved that finding the exact minimum sized subset is an NP-hard problem \[201\]. In practice, knowing the minimum set of pairs may not be as important as the accuracy of determining the permutation mapping. Indeed, the proposed approach determines the permutation mapping if there is sufficient information, and detects the lack of sufficient information when there is not.

Now we evaluate the computational complexity of the proposed chosen-plaintext attack. The first step in the attack procedure is splitting $n$ sources from the $n$ digit expansions in radix $L$ of $MN$ entry locations. The computational complexity of this step is $O(n \cdot MN)$. The second step in the attack procedure is the recombination of $n$ target matrices as radix $L$ values which gives the permuted locations of $0$ to $MN - 1$. The computational complexity of this step is also $O(n \cdot MN)$. As a result, the computational complexity of the proposed attack is $O(n \cdot MN)$. This shows that the proposed cryptanalysis is efficiently achievable by means of a limited number of chosen-plaintexts using a polynomial amount of computation time.

### 6.4 Case Studies – Typical Permutation-Only Image/Video Ciphers

To verify the correctness of the above-discussed chosen-plaintext attack, it was tested on two typical permutation-only image/video ciphers. With respect to this, the recently proposed permutation-only image/video ciphers by Rahman et al. \[187\] and Fu et al. \[188\] are briefly overviewed, respectively.

#### 6.4.1 Rahman et al.’s Encryption Scheme

Rahman et al.’s encryption algorithm contains two parts: a key initializing procedure and a scrambling algorithm. Using a two-dimensional Hénon map \[202\] described in equation \[6.6\] the key initializing procedure provides a binary sequence, which is used as a seed point to run
the scrambling algorithm. The initializing procedure is briefly described as

\[
\{r^n(x, y)\}_{n=0}^{1023} = \{(x_{n+1}, y_{n+1})\} \\
x_{n+1} = 1 + y_n - 1.4x_n^2, \quad y_{n+1} = 0.3x_n \\
\{b_n\}_{n=0}^{1023} = \{\sigma(r^n(x, y))\}_{n=0}^{1023},
\]

where \(\{r^n(x, y)\}_{n=0}^{1023}\) is the chaotic real-valued sequence generated by the Hénon map, \(\sigma\) is the discretisation function, and \(\{b_n\}_{n=0}^{1023}\) is the generated binary sequence. The chaotic binary sequence generated by the Hénon map is then used as the secret key to scramble the position of pixels in a region of interest (ROI).

For a ROI \(R = \{R(i, j)\}_{0 \leq i < M, 0 \leq j < N - 1}\) in an image \(P = \{p(i, j)\}_{0 \leq i < H, 0 \leq j < W - 1}\), the scrambling function employs the following four transformations to scramble \(R\) and map it to \(R' = \{R'(i, j)\}_{0 \leq i < M, 0 \leq j < N - 1}\), where \(H \times W\) represents the size of input image and \(M \times N\) denotes the size of the ROI. The relationships \(M \leq H\) and \(N \leq W\) hold in each image. The first transformation is defined as the mapping \(R' = TRANS1(R)_{r, 292.5^\circ}\), where \(0 \leq j \leq N - 1\). This is designed to rotate each pixel in the \(i\)-th column of the ROI. If \(r = 0\) then the rotation is towards the 292.5° right direction by \(p\) pixels, if \(r = 1\) then the rotation is towards the 22.5° left direction by \(p\) pixels. The second transformation is defined as the mapping \(R' = TRANS2(R)_{r, 112.5^\circ}\), where \(0 \leq i \leq M - 1\). This transformation is designed to rotate each pixel in the \(i\)-th row of the ROI. If \(r = 0\) then the rotation is towards the 112.5° down direction by \(p\) pixels, if \(r = 1\) then the rotation is towards the 112.5° up direction by \(p\) pixels. The third transformation is defined as the mapping \(R' = TRANS3(R)_{r, 202.5^\circ}\), where \(0 \leq k \leq M + N - 2\). This transformation is designed to rotate each pixel at position \((x, y)\) of ROI \(R\) satisfying \(x + y = k\). If \(r = 0\) then the rotation is \(p\) pixels towards the 202.5° upper-right direction, if \(r = 1\) then the rotation is \(p\) pixels towards the 202.5° lower-left direction. The fourth transformation is defined as the mapping \(R' = TRANS4(R)_{r, 292.5^\circ}\), where \(1 - N \leq k \leq M - 1\). This is designed to rotate each pixel at position \((x, y)\) of ROI \(R\) satisfying \(x - y = k\). If \(r = 0\) then the rotation is \(p\) pixels towards the 292.5° upper-left direction, if \(r = 1\) then the rotation is \(p\) pixels towards the 292.5° lower-right direction. Rahman et al. used Algorithm 6.1 to scramble (encrypt) the data in a ROI.

In fact, Rahman et al.’s scrambling algorithm is a permutation-only cipher that encrypts a plain-image by permuting the positions of all pixels in 22.5°, 112.5°, 202.5° and 292.5° degrees towards random directions. Rahman et al.’s scrambling algorithm dissipates the statistical structure of the plain-image into long range statistics. The scrambling algorithm is invertible so the de-scrambling algorithm is possible. Moreover, the scrambling algorithm is influenced by the binary sequence generated by the Hénon map, the dimension of the ROI and the control parameters such as \(n_0, \eta, \xi\) and \(\varepsilon\).
Algorithm 6.1 Rahman et al.’s scrambling algorithm

1: procedure SCRAMBLING($R, M, N, no$)
2: \{Scrambling computes the cipher ROI $R'$ given the input ROI $R$ and the secret key $(M, N, no)$\}
3: \hspace{1em} for $itt \leftarrow 1, no$ do
4: \hspace{2em} $r \leftarrow 2 (2M + 2N - 1) \times (itt - 1)$
5: \hspace{2em} for $bnt \leftarrow 0, 1023$ do
6: \hspace{3em} $p \leftarrow \eta + \xi \times (b_{r+0} \oplus b_{bnt}) + \varepsilon \times (b_{r+1} \oplus b_{bnt})$
7: \hspace{3em} for $j \leftarrow 0, N - 1$ do
8: \hspace{4em} $R1 \leftarrow TRANS1(R)_{b_{r+M+j} \oplus b_{bnt}}^{j,p}$
9: \hspace{3em} end for
10: \hspace{2em} for $i \leftarrow 0, M - 1$ do
11: \hspace{3em} $R2 \leftarrow TRANS2(R1)_{b_{r+i} \oplus b_{bnt}}^{i,p}, 112.5^\circ$
12: \hspace{2em} end for
13: \hspace{2em} for $k \leftarrow 0, M + N - 2$ do
14: \hspace{3em} $R3 \leftarrow TRANS3(R2)_{b_{r+M+N+k} \oplus b_{bnt}}^{k,p}, 202.5^\circ$
15: \hspace{3em} end for
16: \hspace{2em} for $z \leftarrow -(N - 1), M - 1$ do
17: \hspace{3em} $R4 \leftarrow TRANS4(R3)_{b_{r+4M+4N-z} \oplus b_{bnt}}^{z,p}, 292.5^\circ$
18: \hspace{3em} end for
19: \hspace{1em} $R \leftarrow R4$
20: \hspace{1em} end for
21: \hspace{1em} $R' \leftarrow R$
22: \end procedure

6.4.2 Fu et al.’s Encryption Scheme

Fu et al.’s encryption algorithm is a bit-level permutation scheme, which encrypts plain-images in two iterative stages. Firstly, the plain-image is extended into a bit-plane (binary) image, which is constructed by expanding every column of the plain-image into bit-plane columns. An image of size $M \times N$ with 256 colour intensities can be extended to a bit-plane image with size $M \times 8N$. In the first stage, a pseudo-random sequence is generated by a Chebyshev map, ensuring that there is no repetition, and this sequence is interpreted as the permutation mapping. A Chebyshev map is a typical invertible iterated map that generates orthogonal real-valued sequences. The Chebyshev map of degree $D$ ($D = 2, 3, \ldots$) is based on a trigonometric function defined as

$$s_{n+1} = f(s_n) = \cos(D \cos^{-1}(s_n)),$$

where $f : S \rightarrow S, S \in [-1, +1]$. To avoid the harmful effect of transitional procedure, the Chebyshev map is firstly iterated for $N_0$ times, where $N_0$ is a constant. Then, two permutation sequences of length $M$ and $N \times 8$ are generated, which are employed to shuffle the rows and columns of the bit-plane image, respectively. In the second stage, the shuffled bit-plane is firstly divided into eight bit-squares of equal size. Then, each bit-square is shuffled independently with different control parameters by a discretised version of ACM with different control parameters.
The discretised ACM is defined as
\[
\begin{bmatrix}
x_{n+1} \\
y_{n+1}
\end{bmatrix} = \begin{bmatrix} 1 & a \\ b & ab + 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \end{bmatrix} \mod N,
\]
(6.9)

where \( N \) is the number of pixels in one row (or column), \( a \) and \( b \) are control parameters, \( x \) and \( y \) are pixel coordinates, and \( x_n, y_n \in \{0, 1, \ldots, N - 1\} \). The determinant of this map is 1; hence, it is invertible and area-preserving. This stage is iterated \( m \) (\( 1 \leq m \)) rounds. Finally, both stages 1 and 2 are iterated \( n \) times. To construct the cipher image, all the 8 bit-squares are concatenated from left to right and recovered to a pixel-plane. In fact, both stages of Fu et al.’s encryption algorithm can be viewed as a one permutation stage which scrambles the entries of the bit-plane image. As explained by Fu et al. [188], the image translation to a bit-plane image and its inverse are straightforward linear transformations. Therefore, without loss of generality, we assume that Fu et al.’s algorithm encrypts bit-plane images. \((s_0, D, a, b, m, n)\) is the secret key for Fu et al.’s encryption algorithm.

### 6.5 Experiments

According to the proposed cryptanalysis (see Section 6.3), the permutation mapping of the case studies, which were described in Section 6.4, can be easily deduced by \( \lceil \log_L (MN) \rceil \) chosen plain-images. To verify this claim, numerous experiments were performed. Figure 6.2 depicts some of the test images which were used to perform the experiments. These test images were of size \( M \times N = 256 \times 256 \) and \( 512 \times 512 \) with \( L = 256 \) colour intensities. Figure 6.2 also depicts the bit-plane images of the test images. To deduce a unique permutation mapping, the \( \lceil \log_L (MN) \rceil \) chosen plain-images were built based on the proposed coding (see Section 6.3). To verify the breaking performance, the corresponding cipher-images were decrypted with the inferred permutation matrices, and the recovered plain-images were compared with the original test images depicted in Figure 6.2. In the following subsections, the experimental results for breaking Rahman et al.’s and Fu et al.’s encryption algorithms will be given.

#### 6.5.1 Experimental Results for Rahman et al.’s Encryption Algorithm

The test images depicted in Figure 6.2 were encrypted by Rahman et al.’s encryption algorithm using \((x_0, y_0) = (0.45, 0.35), no = 1024, \eta = 1, \xi = 2, \) and \( \varepsilon = 3 \) as the secret key. The corresponding cipher-images are depicted in Figure 6.3. According to the proposed cryptanalysis, to deduce the \( 256 \times 256 \) permutation mapping, the adversary only requires \( \lceil \log_{256}(256 \times 256) \rceil = 2 \) plain-images. In addition, for \( 512 \times 512 \) case, a similar attack procedure requires only \( \lceil \log_{256}(512 \times 512) \rceil = 3 \) plain-images. To deduce a unique permutation
mapping, the plain-images were built based on the proposed coding (see Section 6.3). The chosen plain-images required for cryptanalysis and their corresponding cipher-images are depicted in Figure 6.4. The breaking results of cipher-images #1 and #4 are demonstrated in Figure 6.5.

### 6.5.2 Experimental Results for Fu et al.’s Encryption Algorithm

The bit-plane (binary) images depicted in Figure 6.2 were encrypted by Fu et al.’s encryption algorithm using \( (s_0, D, a, b, m, n) = (0.7, 4, 5, 2, 3, 1) \) as the secret key. The corresponding cipher bit-planes and cipher-images are depicted in Figure 6.6. To deduce the \( 256 \times 2048 \) permutation mapping, the adversary only requires \( \lceil \log_2 (256 \times 2048) \rceil = 19 \) pairs of input/output binary images. For a \( 512 \times 4096 \) case, a similar procedure requires only \( \lceil \log_2 (512 \times 4096) \rceil = 23 \) pairs of input/output binary images.
Chapter 6. Cryptanalysis of Permutation-Only Image Encryption Schemes

![Image of encryption process with 3 stages]

**Figure 6.4:** Required pairs of chosen input/output images (a) with size $256 \times 256$ for finding the permutation matrix of size $256 \times 256$, and (b) with size $512 \times 512$ for finding the permutation matrix of size $512 \times 512$.

![Image of decrypted images]

**Figure 6.5:** Decrypted images of the (a) cipher-image #1 and (b) cipher-image #4.

21 pairs of input/output binary images. To achieve a unique permutation mapping, the input images were built based on the proposed coding. The required pairs of chosen input/output binary images for obtaining the $256 \times 2048$ permutation mapping are depicted in Figure 6.7. Figure 6.7 also depicts the corresponding cipher-images constructed by the output binary images. The breaking results of cipher-images #2 and #5 are demonstrated in Figure 6.8.

### 6.6 Discussion

In this section, we elaborate the advantages of our attack over the chosen-plaintext attacks of [108] and [199]. To this end, we firstly explain the general procedure that is undertaken in a chosen-plaintext attack. To successfully disclose a permutation-only cipher that works on images of size $MN$ with $L$ colour intensities, it is sufficient to input a source image with distinct entries. However, from the practical point of view, constructing a source image with distinct entries may not be feasible, because the set of colour intensities is finite and the number of entry...
locations usually exceeds the number of colour intensities. Therefore, a collection of plain-images, all of which have repeated values, is required to uniquely determine the underlying permutation.

To disclose the underlying permutation mapping, the interest lies in utilising a number of plain-images whose combination using the positional digits, constructs an image with distinct entries. This problem is equivalent to splitting a source image with distinct entries into a number of plain-images whose entries are equal or less than the maximum colour intensity. As explained in Section 6.3, to split the source image, the adversary needs to expand the source entries using $n$ digit expansions in radix $L$ where $n$ digits clearly produce $L^n$ different values. This implies the following relationship for the number $MN$ of entry locations:

$$L^n < MN \leq L^{n+1}. \tag{6.10}$$

The inequalities above indicate that the source entries can be expanded by $O\left(\lceil \log_L (MN) \rceil \right)$ digits, and therefore, the source image can split into $O\left(\lceil \log_L (MN) \rceil \right)$ plain-images. In other words, $O\left(\lceil \log_L (MN) \rceil \right)$ plain-images construct a source image with distinct entries. The expression $O\left(\lceil \log_L (MN) \rceil \right)$ denotes a set of functions $f(L, MN)$, such that, for sufficiently large $L$ and $MN$, there exists a constant coefficient $c$ $(0 < c)$ satisfying $f(L, MN) \leq c \cdot \lceil \log_L (MN) \rceil$. In this inequality, it is certain that $1 \leq c$, because $n$ digits with radix $L$ can produce $L^n$ different colour intensities, and if $n < \lceil \log_L (MN) \rceil$, then $L^n < MN$; and by the pigeonhole principle, at least two entries would get the same values. Therefore, $1 \leq c$.

Following the arguments above, in a known-plaintext attack, in which the plain-images are randomly selected, $c \lceil \log_L (MN) \rceil$ plain-images are required to successfully reconstruct a source image with distinct entries, where $1 \leq c$. In a chosen-plaintext attack, the aim is to
Figure 6.7: Required pairs of chosen input/output bit-plane images with size $256 \times 2048$ for finding the permutation matrix of size $256 \times 2048$.

(a) (b)

Figure 6.8: Decrypted images of the (a) cipher-image #2 and (b) cipher-image #5.
find a procedure with a reduced number of required plain-images. As proved in Section 6.3, a
tight lower bound for the required number of chosen plain-images \( c = 1 \) is achieved when the
\( MN \) source entries are labelled with distinct values \( 0, 1, \ldots, MN - 1 \), and then expanded by
\( \lceil \log_L (MN) \rceil \) digits with radix \( L \).

To ensure the correct retrieval of the permutation mapping by using the least number of cho-
sen plain-images, that is, \( n = \lceil \log_L (MN) \rceil \), an adversary requires a precise and easy method
to construct chosen plain-images. Neither [108] nor [199] proposed exact methods for the con-
struction of the chosen plain-images for a successful chosen-plaintext attack. In [108], Li et al.
provided two rules for the construction of chosen plain-images: (1) the histogram of each chosen
plain-image should be as uniform as possible; and (2) the \( i \)-dimensional \( (2 \leq i \leq n) \) histogram
of any \( i \) chosen plain-images should be as uniform as possible. However, the rules above are
not sufficiently strict, and therefore, they make a great variety of entry arrangements possible
for producing chosen plain-images, which may not lead to a construction of a source image with
distinct entries. Hence, Li et al.’s chosen-plaintext attack may need more plain-images compared
to our chosen-plaintext attack.

For a better comparison, when the number \( n \) of chosen plain-images is \( \lceil \log_L (MN) \rceil \), we
evaluated the performance of the chosen-plaintext attacks with respect to the percentage of cor-
rectly recovered elements of the permutation matrix and the run-time. To this end, we ran 100
independent experiments with distinct \( M, N \) and \( L \). In the performance test, we implemented
the attacks using an un-optimized MATLAB code on a machine with Intel Core i7 2.5 GHz pro-
cessor and 16 GB of installed memory running under Windows 7. The performance statistics for
10 experiments are reported in Table 6.1. The experimental results confirm that compared to the
chosen-plaintext attacks of [108] and [199], the proposed chosen-plaintext attack successfully
recovers the complete permutation mapping with less number of chosen plain-images and less
run-time.

Figure 6.9 depicts the curves for breaking performance of our chosen-plaintext attack and
Li et al.’s cryptanalytic method [108], for a case where \( M = 256, N = 2048, \) and \( L = 2 \).
These curves display the percentage of correctly recovered elements of the permutation matrix,
with respect to the number of chosen plain-images. A comparison between the curves shows
that there is a significant difference between the breaking performances of the attacks, which is
mainly due to Li et al.’s criteria for constructing chosen plain-images. Indeed, Li et al.’s creation
method for chosen plain-images is not precise and it cannot ensure the correct retrieval of the
permutation matrix elements.

Based on the discussions above, the main advantage of the proposed attack over the chosen-
plaintext attacks of [108] and [199] is that it presents a precise method for the construction
of the chosen plain-images which ensures the correct retrieval of the permutation mapping.
In addition, the proposed attack gives a tight lower bound for the number of required chosen
plain-images for a successful chosen-plaintext attack. In other words, while the num-
ber of required plain-images of the chosen-plaintext attacks of [108] and [199] is an order of
TABLE 6.1: Performance test

<table>
<thead>
<tr>
<th>$M$</th>
<th>$N$</th>
<th>$L$</th>
<th>$n$</th>
<th>Correctly recovered elements</th>
<th>Run-time in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Proposed</td>
<td>[108]</td>
</tr>
<tr>
<td>87</td>
<td>296</td>
<td>23</td>
<td>4</td>
<td>100%</td>
<td>45%</td>
</tr>
<tr>
<td>1464</td>
<td>1134</td>
<td>16</td>
<td>6</td>
<td>100%</td>
<td>37%</td>
</tr>
<tr>
<td>378</td>
<td>1202</td>
<td>7</td>
<td>7</td>
<td>100%</td>
<td>37%</td>
</tr>
<tr>
<td>737</td>
<td>1095</td>
<td>28</td>
<td>5</td>
<td>100%</td>
<td>37.5%</td>
</tr>
<tr>
<td>153</td>
<td>1051</td>
<td>4</td>
<td>9</td>
<td>100%</td>
<td>35%</td>
</tr>
<tr>
<td>974</td>
<td>763</td>
<td>16</td>
<td>5</td>
<td>100%</td>
<td>35.5%</td>
</tr>
<tr>
<td>2003</td>
<td>1573</td>
<td>5</td>
<td>10</td>
<td>100%</td>
<td>35%</td>
</tr>
<tr>
<td>815</td>
<td>1042</td>
<td>3</td>
<td>13</td>
<td>100%</td>
<td>34%</td>
</tr>
<tr>
<td>517</td>
<td>716</td>
<td>3</td>
<td>12</td>
<td>100%</td>
<td>33%</td>
</tr>
<tr>
<td>1585</td>
<td>3061</td>
<td>3</td>
<td>15</td>
<td>100%</td>
<td>32%</td>
</tr>
</tbody>
</table>

$\lceil \log_L (MN) \rceil$, that is, $O(\lceil \log_L (MN) \rceil)$, the number of required plain-images of the proposed chosen-plaintext attack is precisely $\lceil \log_L (MN) \rceil$. It is true that for sufficiently large $MN$ and $L$, $\lceil \log_L (MN) \rceil$ (the number of plain-images that our algorithm requires) would grow as fast as any $O(\lceil \log_L (MN) \rceil)$ (the number of plain-images that [108] and [199] require); however, our chosen-plaintext attack is more accurate than Li et al.’s attack, as it provably gives the smallest number for the chosen plain-images. Furthermore, the computational complexity of Li et al.’s attack [108], Li and Lo’s attack [199], and our attack are $O(n (MN)^2)$, $O(n \cdot MN)$ and $O(n \cdot MN)$, respectively, where $n$ denotes the number of chosen plain-images used for a successful chosen-plaintext attack. Although the proposed chosen-plaintext attack and the chosen-plaintext attack of [199] have the same order of computational complexity, that is, $O(n \cdot MN)$, our attack is faster than that of [199], as confirmed by the results of the performance test (see the run-time comparison in Table 6.1).
6.7 On the Security of Permutation–Substitution based Image Encryption Schemes

The plinth of cryptography is built upon the properties of confusion and diffusion as stated by Shannon in [39], and as explained in [203], a large number of encryption schemes are constructed by the combination of substitution and permutation networks to effectively implement Shannon properties. However, any combination of substitution and permutation networks may not provide sufficient security. In this Section, we expand the idea of permutation-only cryptanalysis to analyse more complex cipher structures. It is shown that the one-round combination of a permutation primitive with a substitution primitive such as XOR, as depicted in Figure 6.10, is vulnerable to chosen-plaintext and/or chosen-ciphertext attacks [15]. This is mainly because permutation is not protected from both sides and therefore the permutation–substitution primitives can be analysed using a divide-and-conquer method in one round of operation. This weakness is demonstrated on a case study, that is Pareek et al.’s encryption algorithm.

6.7.1 Case Study

Recently, Pareek et al. [204] proposed an image cipher based on substitution, permutation, and a number of other well-known procedures. The security of the cipher has been evaluated by the authors using statistical analyses. It is true that when a cipher is designed, the security is generally evaluated through statistical analyses [205], [206], [207], and a cryptanalysis procedure, such
Chapter 6. Cryptanalysis of Permutation-Only Image Encryption Schemes

Figure 6.10: One-round combination of a permutation primitive with a substitution primitive.

Algorithm 6.2 Pareek et al.’s mixing step algorithm

1: procedure MIXING_STEP (P, K)
   \{Initial conditions: p(1, 0) = 0 and p(x ≠ 1, 0) = p(x − 1, N)\}
2: \( i = 1 \)
3: for \( x \leftarrow 1, M \) do
4:   for \( y \leftarrow 1, N \) do
5:     \( p(x, y) \leftarrow p(x, y) \oplus p(x, y − 1) \oplus K_i \)
6:   \( i \leftarrow (i \mod 16) + 1 \)
7: end for
8: end for
9: \( M \leftarrow P \)
10: end procedure

as a chosen-ciphertext attack, is not commonly employed to assess image encryption schemes. However, before being applied to practical applications, where security is one of the important requirements, a cipher should be thoroughly analysed making use of all the well-known cryptanalysis procedures, including the chosen-ciphertext analysis. In this paper, we show that the cipher in [204] is broken by a chosen-ciphertext attack.

To elaborate the steps of Pareek et al.’s encryption algorithm, denote by \( P \) the plain-image, \( M \) the mixed image, and \( C \) the cipher-image. Note that each plain-image, mixed image or cipher-image is represented by an \( M \times N \) matrix, where the entry of such a matrix at position \((x, y)\) represents a pixel. For any \( x (1 \leq x \leq M) \), and \( y (1 \leq y \leq N) \), let \( p(x, y) \), \( m(x, y) \) and \( c(x, y) \), be the pixel at the position \((x, y)\) of the plain-image, mixed image and cipher-image, respectively. Pareek et al.’s encryption algorithm is as follows [204]:

1. **Mixing step**: The plain-image \( P \) is taken as the input and is mixed with the secret key \( K = K_1, K_2, \cdots, K_{16} \) using exclusive-or operation as follows. The output of this step is the mixed image \( M \). Algorithm 6.2 describes the mixing step.

2. **Permutation-substitution step**: The mixed image \( M \) is taken as the input and undergoes the following process for 16 rounds. The output of this step is the cipher-image \( C \).
2.1. **Block division**: With respect to the secret key $K = \{K_1, K_2, \ldots, K_{16}\}$, the mixed image is divided into several squared non-overlapping blocks.

2.2. **Permutation**: The pixels of each block are permuted using a zigzag pattern (see [205], for a detailed description of the zigzag pattern).

2.3. **Substitution**: The value of the pixels of each block is altered by masking the pixels with one of their eight adjacent neighbours.

Figure 6.11 demonstrates the block diagram of Pareek et al.’s encryption scheme.

Denote the mixing procedure and its inverse by $Mix(\cdot)$ and $Mix^{-1}(\cdot)$, and the permutation-substitution procedure and its inverse by $PS(\cdot)$ and $PS^{-1}(\cdot)$, respectively. In a nutshell, the encryption procedure is $C = PS(Mix(P))$, and the decryption procedure is $P = Mix^{-1}(PS^{-1}(C))$. In the following, we show that Pareek et al.’s encryption algorithm is vulnerable to chosen-ciphertext attacks. In a chosen-ciphertext attack, the cryptanalyst has temporary access to the decryption machinery, and so is able to construct the plain-image corresponding to a chosen cipher-image. Considering the decryption algorithm, the attack procedure is as follows:

- Given a cipher-image with all pixels being equal to zero ($C = 0$) as the input,
- $PS^{-1}$ process will return the same image ($M = 0$), given the permutation-substitution process by Pareek et al. [204]. Please see Sections 2.2 and 2.3 of [204] for the detail of the permutation-substitution process.
- By the $Mix^{-1}$ process, $P = Mix^{-1}(0)$ is returned as the plain-image. Please see Section 2.4 of [204] for the detail of the mixing process.
- Then, the first 16 elements of the retrieved plain-image $P$ disclose the secret key $K = K_1, K_2, \ldots, K_{16}$. In fact, by $p(x, y) \oplus p(x, y - 1) \oplus K_i = 0$ and using the first row, i.e., $x = 1$, of the retrieved plain-image $P$, we have $K_i = p(1, i) \oplus p(1, i - 1)$, for all $i = 1, 2, \ldots, 16$. With the initial value $p(1, 0) = 0$, we can then get all $K_1, K_2, \ldots, K_{16}$, iteratively. More precisely, from $K_i = p(1, y) \oplus p(1, y - 1)$, substituting $y = 1$ and $p(1, 0) = 0$, we have $K_1 = p(1, 1)$. Then, by substituting $y = 2$, we have $K_2 = p(1, 2) \oplus K_1$. Accordingly, the rest of the key sequence $K_3, K_4, \ldots, K_{16}$ are obtained iteratively. This procedure is summarised in Table 6.2.
Chapter 6. Cryptanalysis of Permutation-Only Image Encryption Schemes

Table 6.2: The first 16 elements of the retrieved plain-image \( P \)

<table>
<thead>
<tr>
<th>Pixel position</th>
<th>Pixel value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( p(1,1) )</td>
<td>( K_1 )</td>
</tr>
<tr>
<td>( p(1,2) )</td>
<td>( K_1 \oplus K_2 )</td>
</tr>
<tr>
<td>( p(1,3) )</td>
<td>( K_3 \oplus K_2 \oplus K_3 )</td>
</tr>
<tr>
<td>( p(1,4) )</td>
<td>( K_1 \oplus K_2 \oplus K_3 \oplus K_4 )</td>
</tr>
<tr>
<td>( p(1,5) )</td>
<td>( K_1 \oplus K_2 \oplus K_3 \oplus K_4 \oplus K_5 )</td>
</tr>
<tr>
<td>( p(1,6) )</td>
<td>( K_1 \oplus K_2 \oplus K_3 \oplus K_4 \oplus K_5 \oplus K_6 )</td>
</tr>
<tr>
<td>( p(1,7) )</td>
<td>( K_1 \oplus K_2 \oplus K_3 \oplus K_4 \oplus K_5 \oplus K_6 \oplus K_7 )</td>
</tr>
<tr>
<td>( p(1,8) )</td>
<td>( K_1 \oplus K_2 \oplus K_3 \oplus K_4 \oplus K_5 \oplus K_6 \oplus K_7 \oplus K_8 )</td>
</tr>
<tr>
<td>( p(1,9) )</td>
<td>( K_1 \oplus K_2 \oplus K_3 \oplus K_4 \oplus K_5 \oplus K_6 \oplus K_7 \oplus K_8 \oplus K_9 )</td>
</tr>
<tr>
<td>( p(1,10) )</td>
<td>( K_1 \oplus K_2 \oplus K_3 \oplus K_4 \oplus K_5 \oplus K_6 \oplus K_7 \oplus K_8 \oplus K_9 \oplus K_{10} )</td>
</tr>
<tr>
<td>( p(1,11) )</td>
<td>( K_1 \oplus K_2 \oplus K_3 \oplus K_4 \oplus K_5 \oplus K_6 \oplus K_7 \oplus K_8 \oplus K_9 \oplus K_{10} \oplus K_{11} )</td>
</tr>
<tr>
<td>( p(1,12) )</td>
<td>( K_1 \oplus K_2 \oplus K_3 \oplus K_4 \oplus K_5 \oplus K_6 \oplus K_7 \oplus K_8 \oplus K_9 \oplus K_{10} \oplus K_{11} \oplus K_{12} )</td>
</tr>
<tr>
<td>( p(1,13) )</td>
<td>( K_1 \oplus K_2 \oplus K_3 \oplus K_4 \oplus K_5 \oplus K_6 \oplus K_7 \oplus K_8 \oplus K_9 \oplus K_{10} \oplus K_{11} \oplus K_{12} \oplus K_{13} )</td>
</tr>
<tr>
<td>( p(1,14) )</td>
<td>( K_1 \oplus K_2 \oplus K_3 \oplus K_4 \oplus K_5 \oplus K_6 \oplus K_7 \oplus K_8 \oplus K_9 \oplus K_{10} \oplus K_{11} \oplus K_{12} \oplus K_{13} \oplus K_{14} )</td>
</tr>
<tr>
<td>( p(1,15) )</td>
<td>( K_1 \oplus K_2 \oplus K_3 \oplus K_4 \oplus K_5 \oplus K_6 \oplus K_7 \oplus K_8 \oplus K_9 \oplus K_{10} \oplus K_{11} \oplus K_{12} \oplus K_{13} \oplus K_{14} \oplus K_{15} )</td>
</tr>
<tr>
<td>( p(1,16) )</td>
<td>( K_1 \oplus K_2 \oplus K_3 \oplus K_4 \oplus K_5 \oplus K_6 \oplus K_7 \oplus K_8 \oplus K_9 \oplus K_{10} \oplus K_{11} \oplus K_{12} \oplus K_{13} \oplus K_{14} \oplus K_{15} \oplus K_{16} )</td>
</tr>
</tbody>
</table>

Pareek et al.’s encryption scheme is vulnerable to chosen-ciphertext attacks due to the following design flaws:

1. Steps 1 and 2 of the encryption algorithm are independent. In other words, any modification made to the procedure of Step 2 can only change the result of this step and has no impact on the outcome of Step 1.

2. Step 2 does not mix the values of pixels with the secret key. The substitution procedure is only dependent on the values of the adjacent neighbours of each plaintext pixel. Actually, in this procedure, the secret key is only used to determine the adjacent neighbours for substitution, but the secret key itself is not used to mask the pixel’s value.

Pareek et al.’s security analysis, which is solely based on statistical tests, did not detect the security flaws. Statistical analyses are commonly used to expose statistical biases of the cipher’s output stream and cannot prove the security. A counterexample for Pareek et al.’s statistical analyses is to give a plain-image constructed by \( P = M \times^{-1}(0) \) as an input to the encryption system. It is clear that \( C = 0 \). Such an image is highly correlated, has a non-uniform histogram, and does not pass the randomness tests [61].

The weaknesses in Pareek et al.’s encryption algorithm can be addressed in the following ways:

1. The mixing step can be included in the permutation-substitution step within the round process, so that the values of pixels will be changed in each round; and
2. The simple substitution in the proposed algorithm can be replaced by a more sophisticated substitution that is capable of masking the plaintexts with a pseudorandom key stream. In this way, the permutation-substitution step not only changes the pixel positions but also changes the pixel values with respect to the secret key.

6.8 Conclusion

In this chapter, we proved that permutation-only image ciphers are completely broken against chosen-plaintext attacks. Based on the proposed attack, the permutation mapping can be easily deduced using an input matrix of size $MN$ whose distinct entries are selected from the $\lceil \log_L(MN) \rceil$ digit expansions in radix $L$ for $0, 1, \ldots, MN - 1$ in respective locations. In a practical attack, the number $n$ of required chosen plain-images to break the permutation-only image encryption algorithm is $\lceil \log_L(MN) \rceil$. It has also been found that the attack complexity is practically small, that is, $O(n \cdot MN)$. This shows that the proposed cryptanalysis is efficiently achievable by means of a limited number of chosen plain-images using a polynomial amount of computation time. Some experiments on a permutation-only image cipher have been performed to validate the performance of the proposed chosen-plaintext attack. Both theoretical and experimental results verified the feasibility of the proposed attack. From the results of this chapter, it is concluded that no better pseudo-random permutations can be realised to offer a higher level of security against plaintext attacks. To offer an acceptable security level against plaintext attacks, the pseudo-random permutations should be updated to a frequency smaller than $\lceil \log_L(MN) \rceil$. In comparison with Li et al.’s and Li and Lo’s plaintext attacks, our cryptanalysis is exact, offering a lower bound on the number of required chosen plain-images and can be achieved in less computation time. The design principle of this new attack is further expanded to break more complex encryption schemes which are constructed by the combination of a permutation primitive with a substitution primitive. It is concluded that a simple combination of a permutation primitive with a substitution primitive may not provide sufficient security and the permutation block should be protected from the input and the output by a substitution (confusion) block.
Conclusions

The research presented in this thesis shed light on the design and cryptanalysis of symmetric encryption schemes for preserving the confidentiality of 3D content. Traditional forms of data security cannot sufficiently address the needs of 3D content security because they mainly process information at the bit-level which is oblivious to the semantics of the information. To overcome this difficulty, several researchers developed special encryption schemes for 3D content. However, as the literature review revealed, these encryption schemes cannot sufficiently address the technical requirements of 3D content and are insecure from the cryptographic point of view. Therefore, to overcome limitations of current techniques in addressing the security challenges of 3D content, we proposed a technical solution by addressing the following requirements:

- **Dimensional and spatial requirement**: A principal requirement for 3D content encryption is to ensure the dimensional and spatial stability of the original content. If such consideration is not taken into account, then encrypted objects may exceed the viewing screen resolution or they may collide with other objects of the virtual world. The literature review indicates that most of the previous encryption schemes cannot maintain such requirement. Although a number of dimension and space preserving encryption schemes (such as [133] and [128]) exist in the literature, they leak important information about the content, because they provide stability in the form of a bounding box. However, from the cryptographic point of view, this method of providing stability is not an appropriate method for the encryption applications as it discloses the maximum dimensional coordinates of the plain point cloud.
Chapter 7. Conclusions

- **Non-isometry of 3D transformations**: Changing the characteristics of 3D geometries alters the conceptual meaning of 3D objects and conveys different information. Hence, it is important to deform such geometries to protect valuable information from the eye of an unauthorised beholder. However, the geometry deformation should be effective, and it must not preserve the distances and angles of the various components of 3D objects. The competitive adversary may attempt to use 3D pattern recognition techniques to find some relationships among the 3D points, and hence, derive the ratio of changes which may lead to the disclosure of hidden patterns. Now, the question is how to effectively deform such complex geometric objects so that the underlying information will remain unrecognisable. The answer to this problem should be sought in the definition of a good cipher. The encryption process must create a completely different output from the original input. In 3D modelling, this suggests the use of non-isometric deformations.

- **Protection of 3D surface geometry from the texture leakage**: Disclosure of the texture image helps a competent adversary in reconstruction of underlying 3D objects. Therefore, the adversary would attempt to identify and locate the boundaries of the protected object within the encrypted texture images. The object boundaries, as well as sharp variations in surface structure, are typically manifested by sharp changes in pixel intensities.

- **Low encryption cost**: One main requirement is to cut down the heavy computational cost associated with the encryption approach. Low cost encryption schemes reduce the load of multimedia processing systems and save software/hardware resources, which is particularly important for real-time processing of 3D content. Using less expensive elementary cryptographic operations, such as random permutations and geometric rotations, 3D data can be effectively encrypted. Since these operations are simple, encryption does not have a high computation cost. However, the challenge is how to achieve a reasonable level of security with such simple operations.

- **Robustness against surface reconstruction attacks**: The study of related works indicates that a major requirement in 3D object encryption is about robustness against surface reconstruction attacks. Any competent adversary would attempt to acquire any knowledge from the point cloud, and therefore, would recognise any hidden pattern to then enable successful reconstruction of the surface. A primary analysis that an adversary may take into consideration is measuring the distance between each pair of points within the ciphertext cloud. The purpose of such analysis would be to cluster any existing pattern inside the ciphertext cloud, such that points in the same cluster have a small distance from one another, while points in different clusters are at a large distance from one another. This may allow partial reconstruction of the surface from an unorganised point cloud.
7.1 Contributions

The previous chapters have detailed sound approaches to overcoming the limitations above. Chapter 4 proposed a technical solution for the problem of 3D object encryption which overcomes the limitations of the current techniques in addressing the confidentiality requirement of 3D objects. The proposed cipher, which is based on a series of random permutations and rotations, is compatible with standard file formats and maintains the semantic requirements of 3D objects, including the dimensional and spatial stability. Since the inverse of permutation and rotation matrices is their transpose, the implementation of the decryption scheme is very efficient. The cipher displaces the plain-points, and thus, deforms the geometry of the 3D object. This deformation is non-isometric. The security of the proposed cipher was convincingly verified by the evidence obtained from cryptanalytic methods and statistical analyses. The result of rigorous cryptanalysis indicated that the proposed encryption scheme is secure against known-plaintext attacks and chosen-plaintext attacks. The performed statistical tests included similarity analysis, plaintext sensitivity analysis, key sensitivity analysis, and spatial randomness analysis. The result of similarity analysis indicated that the plaintext and ciphertext clouds are not only dissimilar but also have dissimilar surfaces. It was shown that the proposed cipher is sensitive to changes of the plaintext and key, and is robust to differential cryptanalysis. The result of the spatial randomness test indicated that there is no appearance of homogeneous zones in the spatial distribution of the cipher point cloud. This shows that the proposed cipher disperses the plain-points randomly into the bounded space. Results of statistical analyses indicated that the proposed encryption scheme is robust against surface reconstruction attacks. In addition to security analysis, the performance of the proposed cipher was tested, and its efficiency for 3D object encryption was validated. Finally, a comparison with existing protection methods showed that the proposed cipher is more effective and has better security despite having the same level of computational complexity.

To overcome the limitations of the current techniques in addressing the confidentiality requirement of texture images, Chapter 5 proposed a technical solution that satisfies the constraints imposed by the structure of texture images, such as large data volume, and the application requirements, such as real-time performance, complexity, and the security level. Compared to the encryption standards, the proposed cipher is relatively lightweight and provides a better encryption performance. It also considerably dissipates the correlation among the entries of the texture image. The key sensitivity analysis showed that even a single bit change in the secret key will result in an entirely different cipher-image. Thus, the original texture image cannot be recovered even though there is a slight difference between the encryption and decryption keys. Furthermore, the proposed texture encryption conceals the shape and boundaries of the underlying 3D object. Therefore, it not only maintains the confidentiality of texture images but also maintains the security of protected 3D models from surface reconstruction attacks using the data provided by the texture images.
Chapter 6 proved that permutation-only image ciphers are completely broken against chosen-plaintext attacks. Based on the proposed attack, the permutation mapping can be easily deduced using an input matrix of size $MN$ whose distinct entries are selected from the $\lceil \log_L (MN) \rceil$ digit expansions in radix $L$ for $0, 1, \ldots, MN - 1$ in respective locations. In a practical attack, the number $n$ of required chosen plain-images to break the permutation-only image encryption algorithm is $\lceil \log_L (MN) \rceil$. It has also been found that the attack complexity is practically small, that is, $O(n \cdot MN)$. This shows that the proposed cryptanalysis is efficiently achievable by means of a limited number of chosen plain-images using a polynomial amount of computation time. Some experiments on a permutation-only image cipher have been performed to validate the performance of the proposed chosen-plaintext attack. Both theoretical and experimental results verified the feasibility of the proposed attack. From the results of this chapter, it was concluded that no better pseudo-random permutations can be realised to offer a higher level of security against plaintext attacks. Also, to offer an acceptable security level against plaintext attacks, the pseudo-random permutations should be updated to a frequency smaller than $\lceil \log_L (MN) \rceil$. This chapter also expanded the idea of permutation-only cryptanalysis for analysing more complex cipher structures. It was shown that a simple combination of a permutation primitive with a substitution primitive may not provide sufficient security and the permutation block should be protected from the input and the output by a substitution block.

7.2 Future Research and Open Problems

The challenge in the development of 3D content ciphers lies in the fact that the design and analysis of such ciphers is presently more art than science. This thesis shed light on some well-established principles for the design and cryptanalysis of 3D content encryption schemes. However, there is a clear need for more scientific formulation of the principles on which the security of such ciphers rests. Also, there is room to further improve the current work to obtain deeper results in both theory and practice. In the following, we highlight some specific problems that need to be further studied.

- The proposed encryption method maintains the location and boundaries of the encrypted content but it is blind to the existence of other nearby objects. Therefore, the plan for future work is to investigate more intelligent methods to control the distortion level of 3D content with respect to its environment.

- Another direction for further research is to investigate the security of the proposed method against more sophisticated techniques in both cryptography and computer graphics, which address 3D reconstruction of synthetic data under antagonistic conditions.

- Supplementary methods should be investigated to extend point cloud based encryption methods to other 3D models, while preserving the application requirements of 3D content.
• In this thesis, we proposed an efficient chosen-plaintext attack for the complete cryptanalysis of permutation-only image encryption schemes. However, our research provides us with an interesting future direction to study the extension of our attack against other types of cipher structures with multiple encryption rounds.
D.1 Plain Texture Images

FIGURE A.1: A number of the used texture images.
D.2 Plain 3D Objects

Figure A.2: A number of the used 3D objects.
Bibliography


