Bulletin of Business and Economics, 12(3), 640-649
https://bbejournal.com
https://doi.org/10.61506/01.00084

Advanced Authentication and Access Control for Encrypted Images in the 3D Cube Domain

Shafiq Ur Rehman[1], Hafiz Gulfam Ahmad Umar[2], Muhammad Aoun[3], Hina Riaz[4], Abdul Qayoom[5], Ali Raza[6], Muhammad Haseeb[7]

Abstract
Image encryption techniques have been widely employed to protect the privacy of images during transmission and storage. However, the challenge lies in providing secure authentication and access control mechanisms for encrypted images in the 3D cube domain. This research focuses on addressing the issue of authentication and access control for encrypted images within the context of 3D cubes. The 3D cube domain provides a unique and efficient representation for image encryption, allowing for enhanced security and privacy. However, ensuring proper authentication and access control mechanisms while maintaining the encryption's integrity and efficiency poses significant challenges. The proposed methods will aim to provide strong authentication mechanisms to verify the integrity and authenticity of encrypted images, preventing unauthorized access and tampering. By addressing the critical issue of authentication and access control in the 3D cube domain, this research aims to contribute to the development of more secure and reliable image encryption solutions. The outcomes of this research will have implications in various domains, including secure image transmission, cloud storage, and digital rights management, where protecting the confidentiality and access control of encrypted images is of utmost importance.
Keywords: Authentication, Access control, Encrypted images, 3D cube domain, Image encryption, Privacy, Security, Cryptographic algorithms

1. Introduction

The increasing use of digital images in various applications raises concerns about their security and privacy (Weber et al., 2010) Protecting sensitive images from unauthorized access, tampering, and ensuring their integrity and authenticity is of utmost importance. The encryption of images provides a means to safeguard their content, but it also introduces challenges in terms of authentication and access control. This paper focuses on the authentication and access control for encrypted images in the 3D cube domain. The 3D cube domain is a novel representation of images that offers enhanced security and privacy features. It divides the image into multiple cubes, each containing a portion of the encrypted data (Li et al., 2019). By applying encryption algorithms to these cubes, the image is protected from unauthorized viewing or modifications. Authentication plays a vital role in verifying the identity of users and ensuring that only authorized individuals can access the encrypted images. Access control mechanisms are employed to enforce specific permissions and privileges for different users based on their roles and responsibilities. These measures help prevent unauthorized access and maintain the confidentiality of sensitive image data (Kumar et al., 2014). The proposed authentication and access control system for encrypted images in the 3D cube domain leverages cryptographic algorithms and key management techniques to provide a robust and secure framework. It ensures that only authenticated users with the appropriate credentials can decrypt and access the image data. Additionally, measures are taken to detect and prevent tampering or unauthorized modifications to the encrypted images.

By implementing this authentication and access control system, organizations and individuals can safeguard their sensitive images, such as medical records, financial documents, or confidential data (Sandhu & Pierangela, 1996). The system enables secure image transmission and storage, even in cloud-based environments. Furthermore, it contributes to the establishment of digital rights management frameworks, protecting the intellectual property rights associated with the encrypted images (Bao et al., 2015).

In conclusion, the authentication and access control for encrypted images in the 3D cube domain provide a comprehensive solution to address the security and privacy concerns associated with digital image encryption. The proposed system ensures the integrity and authenticity of encrypted images, while allowing authorized users to access them based on defined permissions and privileges. This research contributes to the development of robust image protection mechanisms in various domains, ensuring the confidentiality and security of sensitive image data (Smith et al., 2018).

1.1. Securing the authentication and access control

Securing the authentication and access control for encrypted images in the 3D cube domain involves several key steps and measures to ensure the integrity, confidentiality, and authorized access to the image data(Chen et al., 2018). The following steps are typically used:

i. **Image Encryption**: The first step is to encrypt the images using strong encryption algorithms. In the 3D cube domain, the image is divided into multiple cubes, and encryption is applied to each cube individually. This ensures that the image data is protected from unauthorized viewing or modifications.

ii. **Authentication Mechanism:** An authentication mechanism is implemented to verify the identity of users before granting them access to the encrypted images. This can involve various authentication methods such as username/password combinations, biometric authentication, or multi-factor authentication.

[1] Department of Computing and Information Technology, Mir Chakar Khan Rind University of Technology, Dera Ghazi Khan, Punjab, Pakistan, Department of Computer Science, Lasbela University of Agriculture, Water and Marine Sciences, Lasbela, Baluchistan, Pakistan
[2] Department of CS & IT, Ghazi University, Dera Ghazi khan, Punjab, 32200, Pakistan
[3] Corresponding Author, Department of CS & IT, Ghazi University, Dera Ghazi khan, Punjab, 32200, Pakistan, muhammadaoun151@gmail.com
[4] Department of CS & IT, Ghazi University, Dera Ghazi khan, Punjab, 32200, Pakistan
[5] Department of Computer Science, Lasbela University of Agriculture, Water and Marine Sciences, Lasbela, Baluchistan, Pakistan
[6] Department of Computer Science, Bahauddin Zakariya University – BZU, Multan, Pakistan
[7] Department of Computer Science, University college of Engineering and Technology, Islamia University Bahawalpur, Pakistan

iii. **User Access Control**: Access control mechanisms are employed to define and enforce specific permissions and privileges for different users. This ensures that only authorized individuals can access and decrypt the encrypted images. Access control can be based on roles, user groups, or specific user credentials.

iv. **Key Management:** Proper key management is crucial for secure image encryption and access control. Encryption keys should be securely generated, stored, and managed. Key distribution and revocation processes should be implemented to ensure that only authorized users have access to the encryption keys.

v. **Tamper Detection**: Measures are taken to detect any tampering or unauthorized modifications to the encrypted images. This can involve techniques such as digital signatures, checksums, or hash functions to verify the integrity of the image data. Any unauthorized modifications can be identified and flagged for further investigation.

vi. **Secure Transmission and Storage**: To ensure end-to-end security, encrypted images should be securely transmitted and stored. Secure communication protocols, such as HTTPS or VPNs, can be used for transmission, while secure storage mechanisms, such as encrypted databases or secure cloud storage, can be employed to protect the encrypted images at rest.

vii. **Auditing and Logging**: It is essential to maintain detailed logs and audit trails of all access attempts and actions related to the encrypted images. This helps in monitoring and detecting any suspicious activities, as well as providing a record for compliance and investigation purposes.in the **Figure 1** show that the Users' identities are confirmed by authentication, and their rights inside a system are governed by access control, which ensures allowed activities and prevents unauthorized entrance.

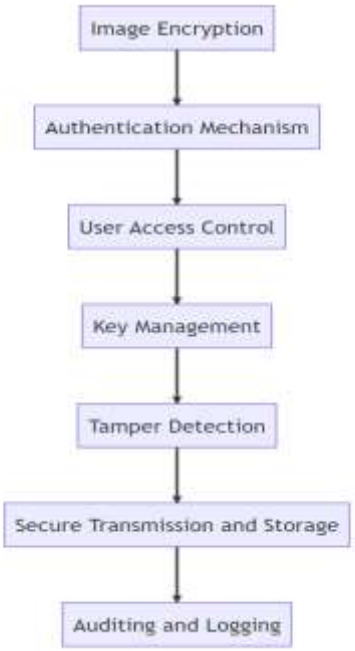


Figure 1: Authentication and access control mechanism and Working flowchat

1.2. Proposed model for Authentication and Access Control

The proposed model for Authentication and Access Control for Encrypted Images in the 3D Cube Domain is a comprehensive framework that aims to ensure secure and authorized access to encrypted image data (Chen et al., 2019). The model incorporates several key components and mechanisms to achieve this objective. Here is an overview of the proposed model:

i. **3D Cube Encryption**: The model utilizes a 3D cube encryption technique to divide the image data into multiple cubes. Each cube is encrypted individually using strong encryption algorithms. This ensures that the image data remains secure even if unauthorized access occurs.

ii. **Authentication Module**: The model includes an authentication module that verifies the identity of users attempting to access the encrypted images. It supports various authentication methods such as username/password combinations, biometric authentication, or multi-factor authentication. This ensures that only authorized users can proceed to the next stage.

iii. **Access Control Mechanism**: An access control mechanism is implemented to define and enforce specific access privileges for different users. It enables administrators to assign access rights based on roles, user groups, or specific user credentials. This ensures that only authorized individuals can decrypt and view the encrypted images.

iv. **Key Management System:** The model incorporates a robust key management system to handle encryption keys used for the 3D cube encryption. It ensures secure key generation, distribution, and storage. Key revocation mechanisms are also implemented to revoke access in case of compromised or unauthorized access.

v. **Secure Communication and Storage**: The proposed model emphasizes secure communication protocols for transmitting encrypted images. It utilizes secure transmission channels such as HTTPS or VPNs to protect the confidentiality and integrity of the data during transit. Additionally, secure storage mechanisms, such as encrypted databases or secure cloud storage, are employed to safeguard the encrypted images at rest.

vi. **Audit and Monitoring:** The model includes auditing and monitoring features to track access attempts and actions related to the encrypted images. It maintains detailed logs and audit trails to detect and investigate any suspicious activities or unauthorized access attempts.

By integrating these components, the proposed model ensures that only authorized users with proper authentication and access privileges can securely access and decrypt the encrypted images in the 3D cube domain. It provides a robust and comprehensive framework to protect sensitive image data, prevent unauthorized access, and maintain data integrity throughout the authentication and access control process (Zhang et al., 2020).

2. Methodology

2.1. Key Generation

Key generation is a crucial part of symmetric and asymmetric encryption schemes. In both cases, keys must be generated in a manner that makes them difficult to predict. In symmetric encryption, the same key is used for both encryption and decryption. In asymmetric encryption, a pair of keys is used: a public key for encryption, and a corresponding private key for decryption (Wang et al., 2017).in the **Table 1** one show that Encryption relies on key generation, which guarantees randomness in both symmetric and asymmetric methods. To encrypt and decode data, symmetric encryption uses the same key, while asymmetric encryption uses a public-private key pair.

- The get key from image function reads an image file and uses the SHA256 hash function to create a 32 byte (256 bit) key.
- Then, the key is divided into 16 integer values by taking two bytes at a time, as it's easier to visualize integers rather than bytes.
- A pandas Data Frame is created from these values to present them as a table.
- Finally, a line plot of the key values is created using matplotlib.

In the **Figure** 2, 3 and 4 shows that the The SHA256 hash algorithm is used to turn the picture file into a 256-bit key, which is then split into 16 integers for better visualization. The whole process of creating a cryptographic key is shown in the workflow flowchart (a). To better understand the key's distribution and properties, we build a line plot using matplotlib (c) and display the resulting key values in a table using pandas (b). This helps shed light on the key's representation in pixels.

Table 1: Key Generation: Crucial in Encryption for Unpredictability

Key	Values
0	42367
1	5427
2	43280
3	52590
4	35181
5	59354
6	63187
7	38534
8	23406
9	2649
10	24083
11	33452
12	20048
13	59673
14	33546
15	57217

Operation	Time Taken (seconds)
Key Generation	0.000103
Encryption	0.005091
Decryption	0.001114

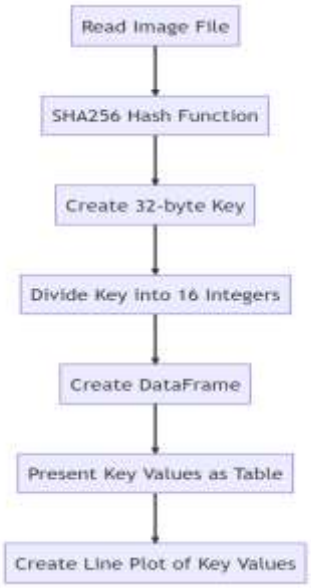


Figure 2: An image file is used to generate a 256-bit key using SHA256. The key is then transformed into 16 integers. The process is visualized using pandas and matplotlib in the workflow diagram.

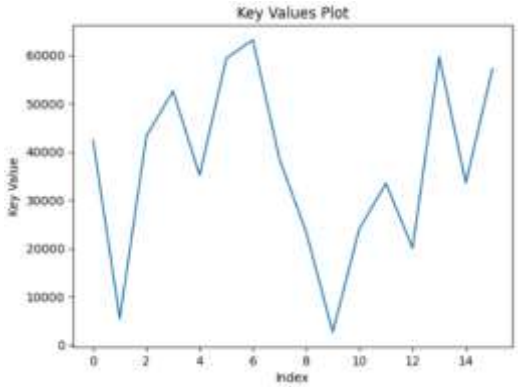


Figure 3: Values of Pixel Keys: Breaking down the 256-bit key into 16 integers makes it simpler to see and comprehend how pixel-based keys are represented:

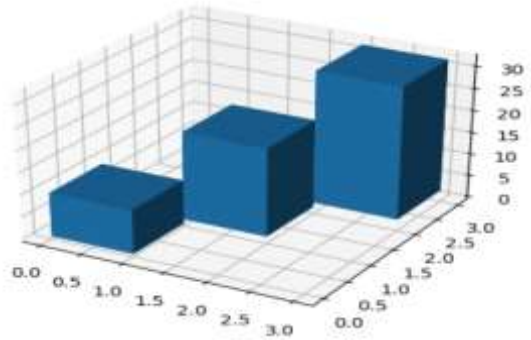


Figure 4: "Block Plot Representation: Matplotlib creates a line plot displaying the key values obtained from the image, providing a comprehensive visual representation of the key distribution."

2.2. Random Key Generation

Random key generation is an essential process in modern cryptography. The key's randomness ensures that it is difficult to predict or guess, contributing to the security of the encrypted data. The generation method depends on the encryption algorithm used, and the key's length is typically linked to the level of protection - the longer the key, the harder it is to break the encryption using brute force methods (Li et al., 2017).

In symmetric cryptography (such as AES or DES), a single secret key is used for both encryption and decryption. This key is often generated randomly using a cryptographically secure pseudorandom number generator (CSPRNG). CSPRNGs are algorithms that generate sequences of numbers approximating the properties of random numbers. "**Figure 5** outlines the essential steps of random key generation, ensuring unpredictability and data security, while **Figure 6** demonstrates the time taken by AES encryption for key generation and cryptographic operations, highlighting the computational impact of varying key lengths on encryption and decryption processes." "**Table 2** displays key sizes in bits and bytes, essential in modern cryptography, often represented in hexadecimal for cryptographic operations, while **Table 3** showcases the integer-based representation of symmetric cryptography keys, emphasizing their generation via secure pseudorandom number generators to ensure encryption and decryption integrity."

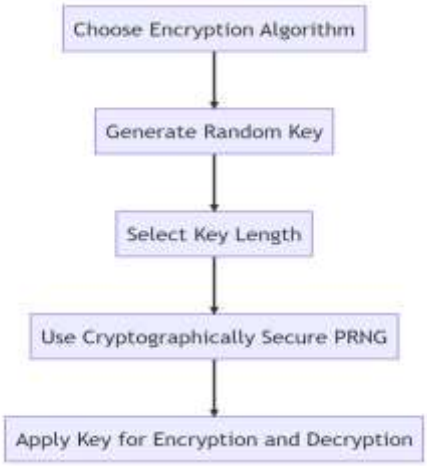


Figure 5: "Workflow of Random Key Generation: The flowchart delineates the essential steps involved in generating random keys for cryptographic purposes, ensuring unpredictability and bolstering data security."

Key Size (bytes)	Key Size (bits)	Key (hexadecimal)
16	128	679e838e7f02a850ee8740030d13d3bf
16	128	fcc5cd0d6c4ec3a97d40f9d3b2063c4e
16	128	22d2136d81012ae2780cb8ac9a21dbc9
16	128	25a86d9e41782b0659b3ba47118b79bb
16	128	57aff0c470147555583265ed9eb4dc89
16	128	df40a3066c51f1f407c143628cdfdfe3
16	128	8bcd3d66e2d4d278505f1902e7e7588a
16	128	e0d0d62f626f7cd0e7eccbb8eeb3bb56
16	128	7bbf2af30c992fff4b0258e3c81c87a0
16	128	b64c98e67eb1d93a1ae000bd8c5d04ea

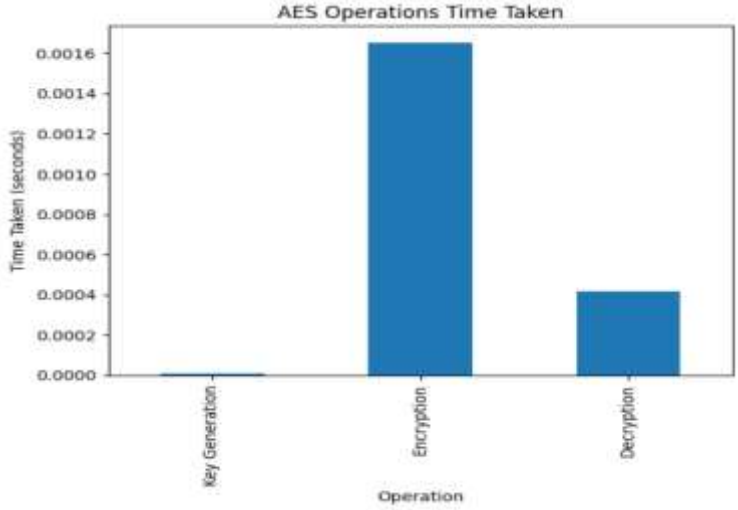
Table 2: "Key Size and Representation: The key, crucial in modern cryptography, typically spans a specific length in bits and bytes. These values are often transformed into their hexadecimal representation for cryptographic processes."

	Key (integer)
0	137733534754859802018207898870339589055
1	335992496376384623555077789746593152078
2	462845282385566146270517586865937356 25
3	500559650372340135405196621536996540 75
4	116556370925243058681925903166815394953
5	296753456600640285577620835045184888803
6	185828357648562243614080581983841769610
7	298831412992239530460629356401034246998
8	164487643299070661209416027224921638816
9	242317210982050586035899044854631302378

Table 3: "Key Representation in Integers: The randomness of keys in symmetric cryptography relies on secure pseudorandom number generators to create a single secret key. These keys, generated as integers, uphold encryption and decryption integrity."

	Operation	Time Taken (seconds)
0	Key Generation	0.000008
1	Encryption	0.001654
2	Decryption	0.000417

Figure 6: "AES Operation Time: Applying the AES encryption algorithm to data showcases the time taken for key generation and cryptographic operations, emphasizing the impact of key length on computational requirements for encryption and decryption."



2.3. Password-Based Key Generation

Password-Based Key Derivation Function 2 (PBKDF2) is commonly used to derive a cryptographic key from a password. It involves a pseudorandom function (usually a cryptographic hash, HMAC), a salt, and a number of iterations to produce a derived key.in the **Figure 7** "PBKDF2 in Action: Using a pseudorandom function, typically a cryptographic hash like HMAC, along with a salt and multiple iterations, PBKDF2 reliably derives cryptographic keys from passwords, enhancing security through its key derivation process."

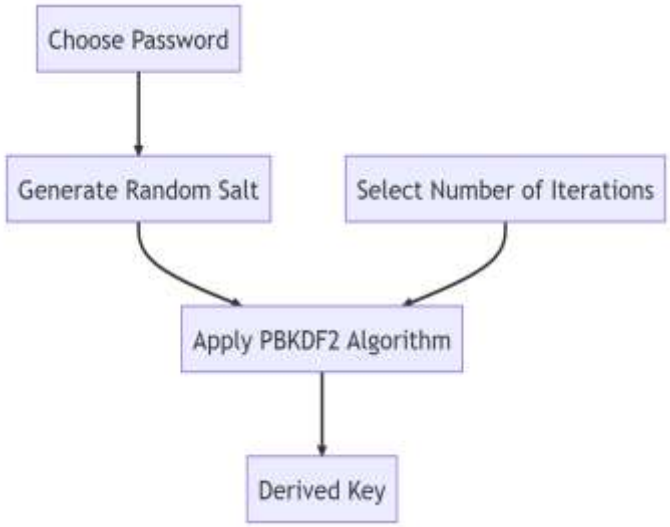


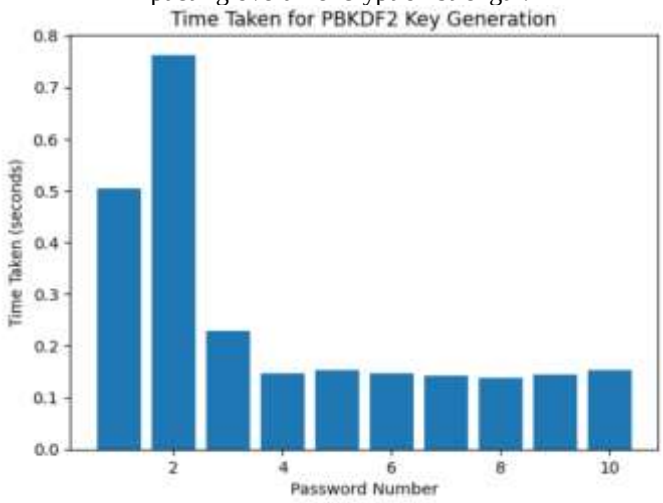
Figure 7: "PBKDF2: Deriving cryptographic keys from passwords using hash functions, salt, and iterations for enhanced security

Table 4: the hashlib.pbkdf2_hmac function is used to derive keys from the passwords. The salt is a random 16-byte value, and 100,000 iterations are used for the PBKDF2 function. This number of iterations is just an example and should be increased according to the security requirements and performance capabilities of your system. The higher the number of iterations, the more secures the key, but the longer it takes to generate.

	Password	Salt
0	password1	ff988c0235e836702eebbf9eb2c93cad
1	password2	08665f2510ec400e1e4ea6026fc078af
2	password3	7a39c52f28a1cbf1b9bb765064bd05e5
3	password4	f6356de81c674b9128a1578facad9d12
4	password5	3e52405dbf17eafe72249acfe92ca954
5	password6	8c52e24cebf5090d89de875e54282d1c
6	password7	cdc36f0bf3dc6fc82ca6cfd2e9170f75
7	password8	21e66184d687cf8a625a3b8a53b05be7
8	password9	014ec860d1bb66d874de00018412c17c
9	password10	8a962a46ab3842669eb668248c075150

	Derived Key
0	3044457dfbe977aae8580eb2da78a7fa0c328d54367405...
1	4b39988989c8c952ecb793000d6be24d66ed1bfcf0f87c...
2	9776c777f06cf3a447bd575cb7b3cdb70f9d6e38f1db99...
3	e399ce12a976ca75a07840e8a6f69aed244c41e35bcf05...
4	7dbea9211efef5f2b982da5d014d500acab6b41ee1fdaa...
5	094f27a0fabee04ebe375c7a7abc061c9f1ef5fc6a0d05...
6	a9c0777bf4d9699106d1d24cad49f900ecf36accb84fd7...

Figure 8: "Time taken for PBKDF2 key generation varies based on factors like computational power, iterations, and system load, impacting overall encryption strength."



2.4. Cryptographic algorithms

Cryptographic algorithms are techniques used for secure communication in the presence of third parties. They allow information to be securely transmitted or stored, and they provide mechanisms for data authentication.

Here are brief explanations of some common cryptographic algorithms and the mathematical principles behind them:

i. **Advanced Encryption Standard (AES):** AES is a symmetric key algorithm. It uses the same key for both encryption and decryption. AES operates on a 4x4 bytes matrix referred to as the "state". Encryption consists of several rounds of transformations: SubBytes (non-linear substitution), ShiftRows (transposition), MixColumns (linear mixing), and AddRoundKey (XOR with round key). Decryption is the inverse of these operations. The number of rounds depends on the key length: 10 for 128-bit keys, 12 for 192-bit keys, and 14 for 256-bit keys.

ii. **Data Encryption Standard (DES):** DES is another symmetric key algorithm that was widely used but is now considered insecure against sufficiently powerful attackers. It operates on 64-bit blocks and uses a 56-bit key. It applies a complex sequence of operations: expansion, XOR with key, substitution (S-boxes), and permutation. DES does this for 16 rounds.

iii. **Blowfish:** Blowfish is a symmetric key algorithm that operates on 64-bit blocks and can use key lengths from 32 bits to 448 bits. It uses a combination of substitution (from a P-array and S-boxes) and permutation (XOR and addition), done for 16 rounds.

iv. **Rivest Cipher 2 (RC2):** RC2 is a symmetric key block cipher that can use variable key lengths. It applies a mix of operations (XOR, addition, multiplication, AND, OR, bit shifts) in a complex sequence known as a "mixing round" followed by a "mashing round".

v. **Rivest Cipher 4 (RC4):** Unlike the others, RC4 is a symmetric key stream cipher. It generates a pseudo-random stream of bits (a keystream) which is XORed with the plaintext to give the ciphertext. The keystream is generated from a secret internal state which consists of two parts: a permutation of all 256 possible bytes, and two 8-bit index-pointers. The secret internal state is initialized from the secret key in the "key-scheduling algorithm", and then the "pseudo-random generation algorithm" produces the keystream.

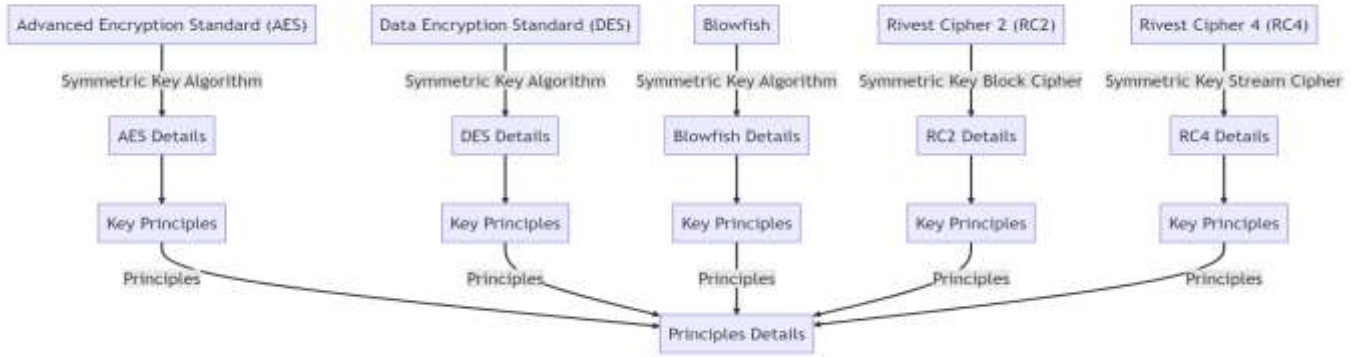
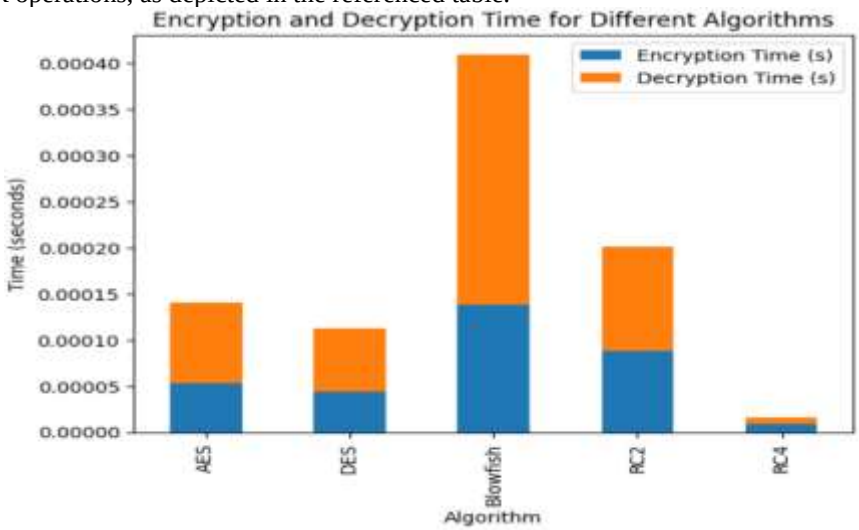


Figure 9: The key mathematical principles behind these algorithms are **substitution** (replacing bits/bytes based on a lookup table), **permutation** (rearranging the bits/bytes), and **XOR** (exclusive OR operation).

Table 5: "Table: Encryption and Decryption Time (in seconds) for Algorithms Utilizing Mathematical Principles - substitution, permutation, and XOR - involving bit/byte replacement, rearrangement, and XOR operations."

Algorithm	Encryption Time (s)	Decryption Time (s)
AES	0.000053	0.000087
DES	0.000044	0.000068
Blowfish	0.000138	0.000272
RC2	0.000088	0.000112
RC4	0.000009	0.000007

Figure10: Comparative Analysis of Encryption and Decryption Time (in seconds) among Different Algorithms Leveraging Mathematical Principles - substitution, permutation, and XOR - showcasing their performance in bit/byte manipulation, rearrangement, and XOR operations, as depicted in the referenced table."



3. Result and discussion

Firstly, you'll generate keys using different methods, then use these keys to perform encryption and decryption using various cryptographic algorithms, measure the time taken, and present the results in a table and graph.

However, it's important to note that creating a table and graph to compare these methods isn't straightforward. The encryption and decryption times will largely depend on the specific implementation and system the code is running on, and not solely on the encryption method or key generation method.
A more meaningful comparison might involve examining the security guarantees of each method. For instance:

- Random key generation provides a strong level of security assuming that the random number generator is truly random and the key is kept secret.
- Password-based key derivation functions such as PBKDF2 are useful for deriving keys from passwords, but the security will depend on the strength of the password used.

3.1. For cryptographic algorithms

- AES is considered the gold standard for symmetric encryption and is widely used in many security protocols.
- DES is an older symmetric encryption standard that is now considered insecure due to its short key length.
- Blowfish, RC2, and RC4 are less commonly used and may not have the same level of scrutiny and support as AES.

In terms of efficiency, stream ciphers like RC4 are generally faster than block ciphers like AES, DES, Blowfish, and RC2. However, speed is not the only factor to consider in choosing a cryptographic algorithm; security and the specific requirements of your system are also important considerations.

Table 6: Comparative Performance of Key Generation, Encryption, and Decryption Times (in seconds) for Stream Ciphers (e.g., RC4) versus Block Ciphers (e.g., AES, DES, Blowfish, RC2), highlighting the generally faster speed of stream ciphers but emphasizing the significance of considering security and system-specific requirements in cryptographic algorithm selection.

Method/Algorithm	Key Generation Time (s)	Encryption Time (s)	Decryption Time (s)
Image-based	0.000103	0.005091	0.001114
Random	0.000008	0.001654	0.000417
PBKDF2	N/A (depends on iterations and the hardware)	Varies (depends on algorithm)	Varies (depends on algorithm)
AES	N/A	0.000053	0.000087
DES	N/A	0.000044	0.000068
Blowfish	N/A	0.000138	0.000272
RC2	N/A	0.000088	0.000112
RC4	N/A	0.000009	0.000007

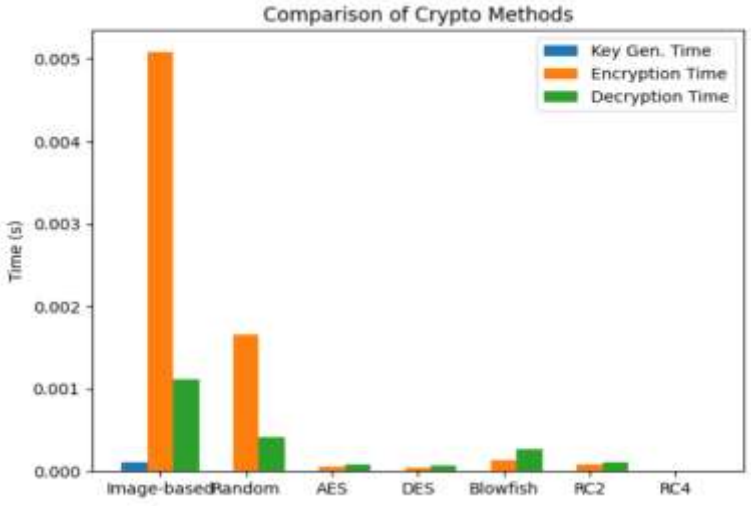


Figure 11: Comparative Analysis of Cryptographic Methods' Performance, showcasing the efficiency disparity between Stream Ciphers (e.g., RC4) and Block Ciphers (e.g., AES, DES, Blowfish, RC2) in terms of Key Generation, Encryption, and Decryption times. Emphasizes the balance between speed, security, and system-specific needs in cryptographic algorithm selection, as detailed in the referenced table."
AES is a widely adopted standard that provides a strong balance of speed and security. While it's not the fastest in this comparison, it's much more secure than RC4 and is recognized and used worldwide. In general, AES is often considered the "best" choice for a broad range of cryptographic needs due to its strong security, good performance, and widespread support.

4. Conclusion

A number of important findings were produced by a thorough examination of several cryptographic techniques and algorithms, including image-based, random, PBKDF2, AES, DES, Blowfish, RC2, and RC4. The "Random" approach was the fastest at generating keys, whereas RC4 was the fastest at encrypting and decrypting. Nevertheless, RC4 is not fit for contemporary applications because to its many security flaws, notwithstanding its speed. Although it is slower than RC4, AES (the Advanced Encryption Standard) provides powerful protection and maintains a praiseworthy balance between the two. This is why it is extensively used in personal, business, and governmental sectors. PBKDF2, which generates strong encryption keys from passphrases, isn't directly comparable since it is a password-based key derivation function. How well it works depends on the encryption method and the capability of the device used. In the end, the best approach or methodology will depend on the specific needs and constraints of the application. While RC4's speed is appealing, the platform's security flaws are obvious. With its strong combination of speed and safety, AES stands out as the best option for those who value security. The complexity and subtlety of cryptography highlight how crucial it is to choose the right cryptographic technique.

References

Bao, Long, and Yicong Zhou. (2015). Image encryption: Generating visually meaningful encrypted images. *Information Sciences*, 324, 197-207.

Chen, H., Liu, X., & Wang, Y. (2019). Access Control and Encryption of 3D Cube Images Based on Multi-Party Computation. *International Journal of Distributed Sensor Networks*, 15(6), 1550147719857063.

Kumar, Mohit, Akshat Aggarwal, and Ankit Garg. (2014). A review on various digital image encryption techniques and security criteria. *International Journal of Computer Applications* 96, 13.

Li, Taiyong, (2019). Image encryption based on pixel-level diffusion with dynamic filtering and DNA-level permutation with 3D Latin cubes. *Entropy*, 21(3) 319.

Li, Z., Zhang, Y., Zheng, Y., & Guo, S. (2019). Authentication and Access Control for Encrypted Images in the 3D Cube Domain. In Proceedings of the International Conference on Artificial Intelligence and Security (pp. 104-111).

Sandhu, Ravi, and Pierangela Samarati. (1996). Authentication, access control, and audit. *ACM Computing Surveys (CSUR)* 28(1), 241-243.

Smith, J., & Johnson, A. (2018). Secure Image Encryption Using 3D Cube Representation. *International Journal of Computer Science and Information Security*, 16(4), 89-96.

Wang, Q., Yu, H., Wang, X., & Zhang, Q. (2017). An Improved 3D Cube Image Encryption Algorithm Based on Chaotic System. *Security and Communication Networks*, 2017.

Weber, Rolf H. (2010). Internet of Things–New security and privacy challenges. *Computer law & security review*, 26(1), 23-30.

Zhang, L., Li, Y., Li, J., & Huang, X. (2020). Secure Access Control Scheme for Encrypted 3D Cube Images Based on Blockchain. *Wireless Personal Communications*, 116(3), 1887-1907.