future internet

MDPI

Article

Secure Video Communication Using Multi-Equation Multi-Key Hybrid Cryptography

Youcef Fouzar [1,*], Ahmed Lakhssassi [1] and Ramakrishna Mundugar [2,*]

[1] Department of Computer Science and Engineering, Université du Québec en Outaouais, Gatineau, QC J8X 3X7, Canada; ahmed.lakhssassi@uqo.ca
[2] Department of Data Science and Computer Applications, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India
* Correspondence: fouy03@uqo.ca (Y.F.); ramakrishna.m@manipal.edu (R.M.)

Abstract: The safeguarding of intellectual property and maintaining privacy for video content are closely linked to the effectiveness of security protocols employed in internet streaming platforms. The inadequate implementation of security measures by content providers has resulted in security breaches within entertainment applications, hence causing a reduction in the client base. This research aimed to enhance the security measures employed for video content by implementing a multi-key approach for encryption and decryption processes. The aforementioned objective was successfully accomplished through the use of hybrid methodologies, the production of dynamic keys, and the implementation of user-attribute-based techniques. The main aim of the study was to improve the security measures associated with the process of generating video material. The proposed methodology integrates a system of mathematical equations and a pseudorandom key within its execution. This novel approach significantly augments the degree of security the encryption mechanism provides. The proposed methodology utilises a set of mathematical equations that are randomly employed to achieve encryption. Using a random selection procedure contributes to the overall enhancement of the system's security. The suggested methodology entails the division of the video into smaller entities known as chunks. Following this, every segment is subjected to encryption using unique keys that are produced dynamically in real-time. The proposed methodology is executed via Android platforms. The transmitter application is tasked with the responsibility of facilitating the streaming of the video content, whereas the receiver application serves the purpose of presenting the video to the user. A careful study was conducted to compare and contrast the suggested method with other similar methods that were already in use. The results of the study strongly support the safety and dependability of the procedure that was made available.

Keywords: asymmetric key cryptography; multi-key encryption; dynamic key generation

Citation: Fouzar, Y.; Lakhssassi, A.; Ramakrishna, M. Secure Video Communication Using Multi-Equation Multi-Key Hybrid Cryptography. *Future Internet* **2023**, *15*, 387. https://doi.org/10.3390/fi15120387

Academic Editor: Carlo Blundo

Received: 29 September 2023
Revised: 11 November 2023
Accepted: 15 November 2023
Published: 29 November 2023

1. Introduction

Online video streaming applications have indeed become essential tools for entertainment purposes. The primary purpose of these applications is to ensure that video content is solely streamed to subscribers. It has been observed that there is a discrepancy between the size of the audience that the video is being streamed to and the number of individuals who have subscribed to the service. Security vulnerability has significantly affected the streaming industry. There are currently several research projects underway that are dedicated to improving security measures and optimising revenue generation.

Based on recent research findings [1], it is evident that the utilisation of over-the-top (OTT) services and video streaming is set to significantly impact the growth of video streaming applications and the subsequent revenue generated within the industry. The results of this survey highlight the critical significance of protecting the copyright of video content and ensuring its security. In recent years, the reductions in the costs of Internet technology and

gadget technology have emerged as significant factors in this particular context. Due to this outcome, online platforms have successfully attracted significant audiences.

The viewer base of over-the-top (OTT) services has witnessed a substantial increase due to the widespread prevalence of illegal file sharing and content piracy. A small fraction of registered users voluntarily discloses their login credentials to individuals within their social circles, such as acquaintances and loved ones. Instances may arise where individuals who have subscribed to certain content engage in replicating it and distributing it to others. An OTT provider's capacity to seek legal remedies is impeded by the abundance of similar disputes, despite the legal ramifications involved.

The utilisation of cryptography techniques [2] plays a crucial role in mitigating the risks associated with unauthorised file sharing and safeguarding against piracy during video content distribution. On the contrary, many investigations have been conducted wherein cryptography has been employed to limit the dissemination of video content. Most approaches in this context rely less heavily on hardware components. Illegal file sharing and streaming inevitably occur due to certain factors.

Symmetric and asymmetric key cryptography are two prominent methods employed to ensure robust security in the communication process. Symmetric key cryptography, also referred to as secret key cryptography, is the predominant cryptographic technique employed in multimedia communication [3–5]. The Advanced Encryption Standard (AES) is widely recognised as the most effective and extensively utilised form of symmetric cryptography [6]. Websites and web browsers utilise the 128-bit Advanced Encryption Standard (AES) algorithm to ensure robust security for online communication. The utilisation of the Secure Real-time Transport Protocol (SRTP) in this particular procedure has presented challenges in terms of key management throughout its entirety. As the network's decentralisation increases, the complexity of addressing the key management problem also intensifies. The task of successfully splitting the key or algorithms utilised in various internet safety enhancement methods is widely regarded as an insurmountable challenge. Due to the implications of the factors mentioned above, there is a notable surge in ongoing research endeavours aimed at advancing asymmetric key cryptography methodologies [7,8]. Asymmetric cryptography, also known as public-key cryptography, has emerged as a significant solution to the challenges posed by symmetric cryptography. Prominent methods in this field include Rivest–Shamir–Adleman (RSA) [9] and Elliptic Curve Cryptography (ECC) [10–12]. These methods have been extensively studied and are currently gaining recognition for their effectiveness in addressing the limitations of symmetric cryptography. Figure 1 illustrates the application of asymmetric key cryptography in the context of video transmission.

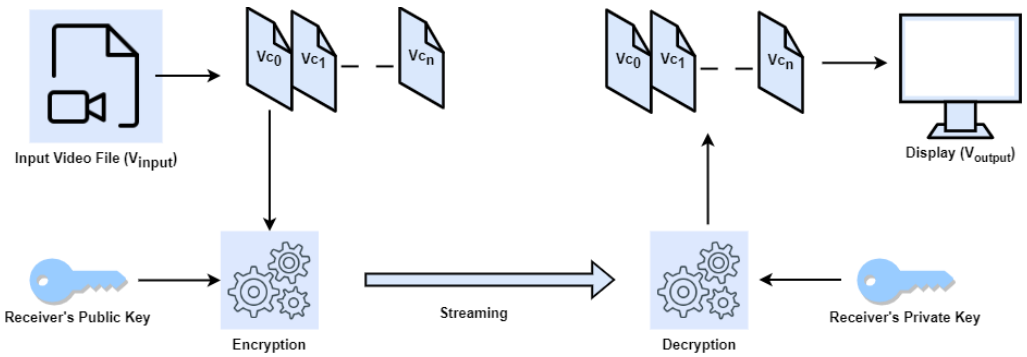

Figure 1. Using asymmetric key cryptography for secure video communication.

In order to optimise the real-time transfer of substantial data volumes, particularly videos, video streaming programs commonly employ a technique known as data segmentation. This involves dividing the data into smaller, more manageable segments. Traditional encryption and decryption techniques predominantly depend on symmetric key cryptography, which employs the same key for both encryption and decryption processes. However, it is important to note that security vulnerabilities can emerge due to the manner

in which keys are exchanged between parties involved in the communication process. Hence, implementing asymmetric key cryptography techniques can significantly enhance the security level for online video transmission. The utilisation of a dynamic key generation technique is imperative for this approach due to the inherent static nature of the keys employed. This study presents a novel approach to encrypting video segments using asymmetric key cryptography principles. This research involves the design and development of a novel equation-based multi-key encryption technique and a method for generating video-attribute-based decryption keys.

Multi-key cryptography provides a resilient method for ensuring security by utilising many keys and participants in cryptography processes. The primary benefit of this approach is its enhanced security, as it becomes considerably more challenging for attackers to break numerous keys simultaneously. The utilisation of this approach facilitates the dispersion of trust across involved actors, rendering it applicable in the context of safe inter-organisational interactions. Moreover, it streamlines the process of managing cryptography keys, guarantees the preservation of anonymity in collaborative computations, and facilitates the use of threshold cryptography for activities that necessitate collective agreement. Multi-key cryptography enhances the overall integrity of a system.

Dynamic key generation is a key part of cryptography. By making private keys on the fly, it improves security. This method increases security by ensuring that keys last only a short time and are changed often. This makes it hard for attackers to take advantage of security holes. Notably, dynamic key creation is based on the idea of "perfect forward secrecy", which means that each session key protects both past and future communications, even if the key is compromised. Cryptography systems can react to changing security needs and operational dynamics by making keys on the fly. This makes key compromises less dangerous and stops attacks, like replay attacks and cryptanalysis, from happening. This method is not only in line with the rules for key management, but it also protects data security, privacy, and compliance in a digital world that is always changing.

The primary objective of this research work is to develop a robust cryptography technique that incorporates dynamic key generation, the utilisation of multiple keys, and the application of asymmetric cryptography methodology. The security of our earlier work [13] is improved by this effort, as Figure 2 illustrates. The previous work employed a hybrid multi-key cryptography technique. In the previous work, the Elliptic Curve Cryptography (ECC) equation was used to generate distinct keys for each video chunk, which were subsequently utilised for encryption. In this study, we aim to optimise the encryption complexity by leveraging a series of mathematical equations. In addition, user attributes are considered alongside the MAC address. The salient characteristics of the proposed methodology are outlined below:

- Hybrid multi-key: The proposed technique encrypts the video segments using RSA and AES. In this case, the video data and its accompanying metadata are encrypted using different passwords.
- Dynamic key generation: A dynamic and automatic key generation technique based on equations has been implemented. The algorithm can produce distinct keys for each video chunk.
- Equation set: A set of equations is used in dynamic key generation. In [13], we have used only one equation for every chunk of video data. Here, we are increasing the complexity of the cryptography technique using the equation set. Here, each video chunk will use a different equation from the set.

The proposed model has been developed and evaluated in a real-world environment. Subsequently, a comprehensive analysis of its performance has been carried out, focusing on the delay parameters across a wide range of circumstances. Given the uniqueness and novelty of the concept, we have opted to utilise our previously validated research as a basis for conducting a comparative study. The research discusses the impact of randomness on computational complexities and provides an understanding of its influence on visual communication. The primary goal is to achieve real-time streaming while upholding a

robust security framework. The Android platform has been carefully considered for both the implementation and streaming of the video sequence.

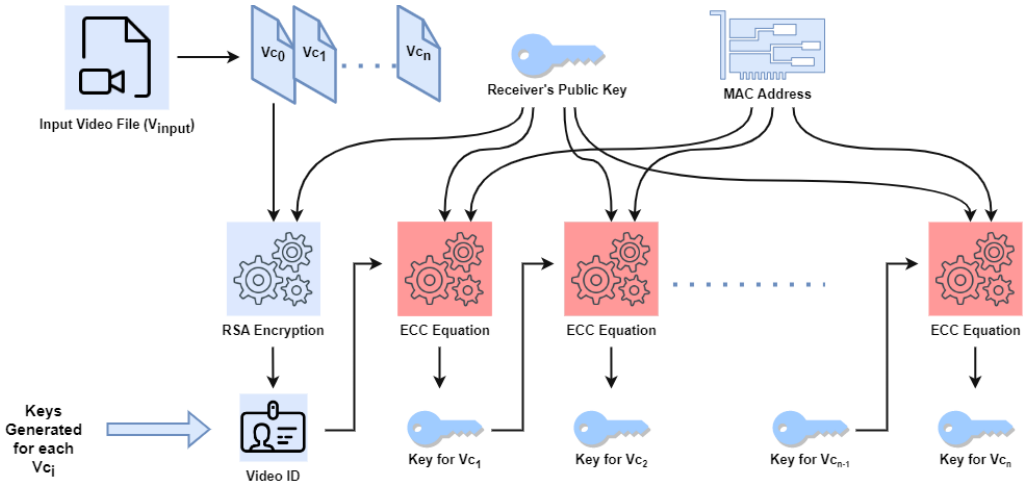

Figure 2. Multi-key-based hybrid cryptography technique [13].

The rest of the paper is organised as follows: Section 2 discusses the related works. Section 3 presents the proposed mechanism. the evaluation and numerical results of the algorithm are detailed in Section 4. Finally, the conclusions drawn are described in Section 5.

2. Related Work

China et al. [14] offer a collection of tools for creating and using keys in the context of both an external world and an identity-based environment. This unique naming scheme, which only supports a few situations at the moment, solves the problem of key revocation (e.g., expire by date, expire by year, and valid for year). Incorporating X.509 standards into identity-based cryptography (IBC) in this implementation gives a high level of interoperability compared to previous efforts on hybrid PKI-IBC systems. The minimal services can be combined with already-existing tools like the Enterprise Java Bean Certified Authority (EJBCA).

Obaidat et al. [15] suggested a more secure way of authentication. This approach lowers the number of vulnerabilities that are a direct product of the process while maintaining the current structure of accepted paradigms. Additionally, it does not raise the level of accountability placed on users or server administrators. The suggested approach uses a hybrid, layered encryption strategy and a two-step verification procedure. Without increasing vulnerabilities for other attacks, such as brute-force attacks, this combination provides dynamic protection against interception-based cyberattacks, such as replay or MiTM attacks.

The capabilities of encryption for providing security in distributed storage were introduced in [16,17]. This was accomplished by investigating standard encryption methods like AES, ECC, and RSA. The problem of identifying an efficient and secure encryption method was, in any event, resolved by this research with regard to the variations in the exhibition of these processes; while some encryption techniques can guarantee security, they require a long time to encode and decrypt data. However, while various methods may provide effective encryption, they still suffer from the negative impacts of the requirement for security.

A two-level cryptographic method and a model for enhancing information security in cloud computing were proposed in [18–20]. In order to increase the security of the information against intrusion, the model uses symmetric and asymmetrical encryption computation (AES and ECC). By preventing them from accessing accurate information, it strengthens information protection, legitimacy, and processing times required for crypto-

graphic operations. This increases the degree of customer confidence in cloud computing and speeds up the use of ECC's more constrained keys in cryptographic interactions.

An equation-based cryptography protocol is discussed by Micciancio et al. [21]. Giacon et al. [22] have proposed a hybrid method that considers a multi-user condition and provides security.

Chaudhari et al. [23] reviewed attribute-based encryption techniques. Attribute-based encryption (ABE) is a type of public key encryption that allows users to encode and rewrite communications based on the user's approved attributes. The scale of the ciphertext is directly proportionate to the number of attributes associated with it, and thus, the encryption duration is proportional to the number of attributes used throughout encryption. Several reasonable ABE versions use one pairing procedure per attribute throughout secret writing.

Niu et al. [24] suggested an attribute-based searchable encryption method with blockchain-verified ciphertext. The keyword, symmetric key, and file are encrypted using the public key method, attribute-based encryption, and symmetric key. The blockchain stores the term index, and the cloud server stores the symmetric key and file ciphertext. The authority centre handles user attribute removal using proxy re-encryption technology when the user's attributes or ciphertext access structure changes.

Wang et al. [25] proposed a new attribute-based encryption scheme that used a fixed-size key. The access policy in this work can be expressed as any monotone access structure. The size of the ciphertext is independent of the number of ciphertext attributes, and the number of bilinear coupling evaluations is reduced to a constant value. Using the general Diffie–Hellman exponent assumption, they establish that the scheme is semantically secure in the selective-set model.

Hohenberger et al. [26] have proposed attribute-based encryption and decryption algorithms. This study focuses on expressive systems without system-wide limits or restrictions, such as limiting the number of attributes used in a ciphertext or a private key. This is the first key-policy ABE system that allows decryption using a constant number of pairings.

The study conducted on the related work helped to understand the use of RSA in hybrid methods. The utilisation of RSA in hybrid cryptography presents various benefits. The key exchange and distribution features of RSA are particularly notable because of its ability to safely facilitate the exchange of symmetric keys between communication parties. This enables the subsequent encryption of data using the shared key. This solution achieves a harmonious equilibrium between security and efficiency by leveraging RSA's asymmetric encryption for secure key exchange while simultaneously capitalising on quicker symmetric encryption algorithms for expedited data transmission. Moreover, the adaptability of RSA enables it to effectively execute diverse cryptographic operations, including encryption and digital signatures, rendering it highly adaptable to a wide range of security demands. Hybrid cryptography is a method that leverages the benefits of both RSA encryption and symmetric encryption to achieve enhanced performance and solid security measures for safeguarding data and facilitating communication.

From this literature, it is observed that the hybrid model is a suitable method for enhancing the robustness of secured data. The previous work [13] concentrated on combining AES and ECC to achieve hybrid and multi-key encryption techniques. However, there is notable work on user-attribute-based key generation techniques. Furthermore, from the literature review, it is evident that there has been no work exploring multi-equation-based key generation techniques to achieve higher security. Hence, this work aims to develop a hybrid multi-key encryption technique.

3. Proposed Method

In this study, we have proposed a novel multi-equation and user-attribute-based cryptography technique to improve security in video communication. The proposed method uses RSA and AES methods to achieve hybrid encryption and decryption. Then, we use a set of equations and randomisation to improve the complexity and, hence, better security.

3.1. Problem Statement

Within the context of a video streaming application, video is sent from a media server to client devices on an as-needed basis. Securing digital video content includes various security measures, such as conditional access, user authentication, content copy control, and video content tracking. In most cases, cryptography procedures are used to carry out the implementation of these security measures. Nevertheless, researchers face a difficult task in their pursuit of a comprehensive answer to the problem of digital video security.

In the realm of cryptography, many studies have been carried out to investigate the advantages of utilising asymmetric key cryptography methods to overcome difficulties associated with key management. The currently available methods do not support dynamic or automatic multi-key techniques aimed at improving video communication applications' security. Consequently, a method of key management that is both automatic and dynamic is required. As a consequence of this, an RSA- and AES-based multi-key encryption technique will serve as the primary focus of this study.

3.2. Proposed Algorithms

A novel multi-equation-based multi-key crypto technique is proposed in this study. We have improved upon the method proposed in [13] using multi-equation and user attributes. The previous method uses RSA, AES, and ECC equations to encrypt video chunks. Here, each video chunk uses an ECC equation to generate the key for chunks. (We recommend the readers to read the article for a better understanding). The proposed method has the following features:

- Achieves security using content metadata and receiver's details.
- RSA, AES, and equations make the method hybrid.
- Increases the complexity of the security using randomisation.
- Improves security using user attributes.
- Simplifies key management using dynamic and automatic generators.

Figure 3 shows the proposed cryptography techniques and the usage of AES, RSA, multi-equations, and user attributes in the process. In this process, the multi-key generation module are detailed and discussed.

The key generation process begins with the input of the video data. The video is divided into multiple 1 MB size video chunks. Here, it is considered that the video chunks are of an equal size to 1 MB. However, while generating the chunk, we have ensured that the chunk will have the full frame. The video chunks are referred to as Vc_i, where i is a sequence number starting from 0 to n. Hence, Vc_0 refers to the first video chunk and so on. The sender device keeps track of every receiver and their characteristics, including the user's public key R_{Pkey} and user attributes U_{attr} (MAC address, and so forth). The process of creating the video ID kicks off the video streaming. Later, the chunks are encrypted independently using various keys.

In Algorithm 1, each step of the proposed method is broken down into its parts. Generating video segments Vc_i that are all the same size helps keep the algorithm's complexity under control. The public key of the recipient R_{Pkey} is utilised, along with the first 16 bytes of the video data, to generate the video ID V_{ID}. Initial video bytes are available in this 16-byte block, and that information is regarded as being exclusive to each video chunk. As a result, the video ID V_{ID} generated in this phase is one of a kind, and it is this video ID utilised for the subsequent encryption of chunks. The video ID V_{ID} is transformed into the base64 string format to be saved and used within the program.

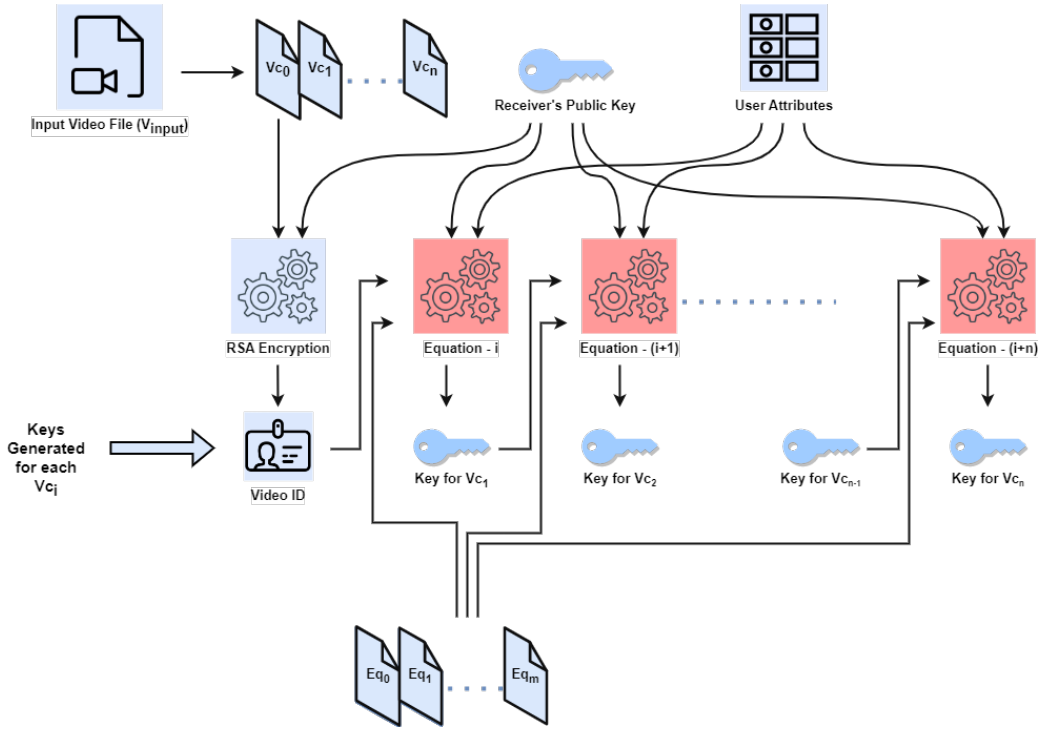

Figure 3. Proposed multi-equation-based hybrid multi-Key cryptography method.

Algorithm 1 Encryption flow

1: Input video file V_{input}
2: Generate video chunks Vc_i, $i = 0$ to n from V_{input}
3: Fetch the receiver's public key of the R_{Pkey}
4: Fetch receiver's attributes U_{attr}
5: Generate V_{ID} using first chunk Vc_0
6: Store V_{ID} in a temporary file
7: Encrypt first chunk Vc_0 using RSA
8: Read index value j from the receiver agreement base
9: Generate $Key_a \leftarrow EQ_j\{V_{ID}, U_{attr}\}$
10: Encrypt Vc_1 using Key_a and AES
11: **for** i:=2 to n **do**
12: Generate $Key_a \leftarrow EQ_j\{Key_a, U_{attr}\}$
13: Encrypt Vc_i using Key_a and AES.
14: **end for**

As shown in Algorithm 1, the video ID V_{ID} is used for generating the key. This key is used by the first video chunk only. The video chunk is encrypted using AES and streamed to the client.

The set of equations comprises an extensive assemblage of mathematical expressions that communicate a diverse array of information. The set of equations encompasses a range of mathematical expressions, including quadratic equations, which are characterised by second-degree polynomials, linear equations that can be graphically represented as straight lines, radical equations that incorporate radicals, exponential equations that involve exponents, and rational equations that incorporate fractions. Every category of equation possesses distinct characteristics and approaches for resolution, rendering them valuable in diverse circumstances. The equation EQ_j is the equation that is chosen from the set. The initial value for j will be decided in the process of user creation for the communication. The value of j is maintained by both the sender and receiver, and a later initial value is used in the encryption and decryption of the video chunks. A numbering system that follows a progression is used to organise the equations in a file. The movie is encrypted using

a variety of equations at each chunk level. An integer chosen at registration is used to determine which set of the equation serves as the beginning of the sequence. The attributes related to user registration are identified and shared between the media server and the user. This is used in the encryption process. Hence, each user will have a different set of equations for the processing.

The determined equation set is put to use to produce the key for the encryption process. As a result, each chunk is assigned a unique key that is produced by a distinctive equation. The complexity of the suggested algorithm will increase as a result of this. As demonstrated in Algorithm 1, the equation is utilised in conjunction with the key generated earlier for the preceding chunk. The algorithm makes use of many different user attributes, such as the MAC address, user name, contact information, and so on, in order to make it more difficult to use.

Following the generation of a unique key for each video chunk, the Advanced Encryption Standard (AES) algorithm is used to encrypt the video chunks. Within the context of this multi-key and hybrid method, the suggested method does not share any of the keys. The key for video communication is generated on the run, as detailed in the algorithm; as a result, the highest possible level of security can be attained. The fact that brute-force attacks can only compromise one chuck at a time does not compromise the security of the remaining video data. This is one of the most important aspects of the suggested method.

Similarly, the algorithm shown in Algorithm 2 demonstrates the steps involved in the decryption process. In our proposed method, the process of decryption remains the same as the encryption steps.

Algorithm 2 Decryption flow

1: Input encrypted Vc_0
2: Read receiver's public key of the R_{Pkey}
3: Read receiver's attributes U_{attr}
4: Decrypt first encrypted chunk Vc_0 using RSA
5: Generate V_{ID} using first chunk Vc_0
6: Store V_{ID} in a temporary file
7: Read index value j from the agreement base
8: Generate $Key_a \leftarrow EQ_j\{V_{ID}, U_{attr}\}$
9: Decrypt Vc_1 using Key_a and AES
10: **for** i:=2 to n **do**
11: Generate $Key_a \leftarrow EQ_j\{Key_a, U_{attr}\}$
12: Decrypt Vc_i using Key_a and AES.
13: **end for**

Case Study:

The key is created for each chunk as follows:

Initially, the video ID is generated and used for generating the subsequent keys. To generate V_{ID}, the process uses a public key R_{Pkey} and a user attribute U_{attr}. Later, the equation set is used for key generation. Assume the following equation has been used to generate the key for the first video chunk Vc_1.

$$y = ax^2 + bx + c \tag{1}$$

The proposed method considers V_{ID} as a and U_{attr} as b in Equation (Equation (1)). In the proposed method, a value for c will be generated at the time of user registration. The same value of c will be used throughout the process. Hence, the modified equation is as follows:

$$Key_a = V_{ID} * x^2 + U_{attr} * x + c \tag{2}$$

The first video chunk uses Equation (2) to generate the key. The second chunk will obtain a different equation, which is assumed here as Equation (3). The algorithm considers previously computed Key_a instead of V_{ID}, as shown in Equation (4). This process continues for all the remaining chucks in the video.

$$y = \frac{-ax - c}{b} \tag{3}$$

$$key_a = \frac{-Key_a * x - c}{U_{attr}} \tag{4}$$

The key key_a is generated using Equations (2) and (4) specified in the encryption and decryption method. Given the consideration of the 128-bit key size for the Advanced Encryption Standard (AES), it is necessary to pad additional bits to the variable key_a in order to achieve the desired size.

Multi-equation multi-key hybrid cryptography is a versatile approach that combines symmetric and asymmetric encryption to provide a robust security framework for data protection. It offers a potent solution for safeguarding data confidentiality by using asymmetric encryption to securely exchange a symmetric key, which is then employed for efficient bulk data encryption. This approach not only enhances data privacy but also mitigates key management challenges. By allowing multiple recipients to encrypt data using their respective public keys, multi-key hybrid cryptography enables scalable and secure communication, ensuring that each recipient can only access data intended for them. Moreover, it provides resistance against common cryptographic attacks due to the use of strong, randomly generated symmetric keys. However, it is imperative to emphasise that the security of this scheme heavily relies on meticulous key management practices and the protection of private keys. Additionally, the video-chunk-level encryption using different equations enhances the robustness of the proposed model.

While multi-key hybrid cryptography boasts several strengths, it is not without its challenges. Complex key management, especially when dealing with a multitude of recipients, can introduce overhead and potential vulnerabilities if not handled diligently. Furthermore, achieving data integrity may require additional cryptographic mechanisms, such as digital signatures or hash functions, which can increase computational costs. Nonetheless, with proper implementation, secure key handling, and adherence to best practices, multi-key hybrid cryptography remains a potent tool in the arsenal of cryptographic techniques, addressing various security requirements and ensuring the confidentiality and privacy of sensitive data in modern communication systems.

In theory, both the AES and the RSA encryption algorithms possess vulnerabilities that make them susceptible to brute-force attacks. The method under consideration employs a key size of 128 bits, which is relatively shorter compared to other key sizes. Consequently, there exists a potential for the key to be learned over an extended period. The proposed model incorporates a multitude of parameters in order to derive the key, thereby significantly augmenting the complexity involved and subsequently diminishing the potential for a successful brute-force attack. The sharing of basic details is facilitated by using static user attributes, such as the Media Access Control (MAC) address, among other methods.

4. Experimentation

The proposed cryptography method is implemented using the mobile platform. The encryption and decryption procedures, as well as video streaming, are implemented using an Android-based application.

When implementing the RSA and AES algorithms in the Android platform, choosing the appropriate security parameters is crucial. In the context of RSA, a key size of 2048 bits is selected, which is considered to be a robust choice for ensuring the security of the encryption algorithm. Additionally, a secure padding scheme known as Optimal Asymmetric Encryption Padding (OAEP) is employed (we have implemented the method discussed in [27]).

This padding scheme enhances the security of RSA by introducing additional randomness and mitigating potential vulnerabilities. When considering the Advanced Encryption Standard (AES), selecting an appropriate key size is crucial based on the desired level of security. In this case, a key size of 128 bits is chosen. Additionally, the AES-GCM mode of operation is selected, which provides both confidentiality and integrity protection. To ensure the effectiveness of AES-GCM, an initialisation vector (IV) is utilised. The IV is generated using the proposed method for generating keys. Firstly, the Android development environment needs to be established in order to proceed with the required tasks. Once the environment is set up, the Android Keystore is utilised for secure key storage. To ensure data security, encryption and decryption functionalities are implemented. The KeyPairGenerator class is employed for RSA operations, while the Cypher class is utilised for AES operations.

The sender devices keep the video material and video metadata for streaming. The sender software launches when a receiver asks to stream a video file. The sender, an authorised video distributor, confirms the recipient's identity before sending the video material to them. The receiver data are maintained in the device. During the sign-up procedure, information is gathered about the recipient. The sender's database is retrieved and filled with all data, including the user name, password, public key, and user attributes, like the MAC address.

Decryption and video replay are the receiver's primary responsibilities. The proposed module gets encrypted video chunks and decrypts them. The receiver's database is used to acquire the required keys. Therefore, in this technique, no key exchange takes place. The sender's public and recipient's private keys are used in the decoding. The program uses an RSA implementation to get the first video segments. The implemented application then shows the video data on the device's display unit. As a result, there is no need to save the obtained video data.

A database must be set up for the applications to keep the video metadata and information about the receiver. On the sending side, the database keeps track of the following: information about the recipient, such as their username, the MAC address of the device, the public key, and other information that is unique to their account. In the same way, the sender's public key and the video session's metadata are saved in the receiver's database.

The suggested method uses a collection of equations to generate the dynamic key automatically. The suggested module uses the trap-door mechanism, a well-known feature of the equation-based method. The module considers the receiver's public key in addition to the previously generated key that was discussed in the previous Section 3, as well as a set of user attributes.

The sender-side module creates the segments after reading the video file. In this case, the FFMpeg module created the video chunks. The movie has been divided into a unit size. Regardless of the chunk size, which can be any size, the module generates the chunks to the nearest complete frame.

In this section, the proposed cryptography method was assessed using a number of parameters, including the time it took to generate each key, the time it took to encrypt files with different file sizes and chunk sizes, the number of keys generated, and the time it took from the generation of keys to the completion of encryption. This also applies to decoding.

4.1. Delay for Splitting File to Chunks

This metric was used to monitor the amount of time that the program required to generate the video chunks from the video file. Implementing video encryption based on chunks is one of the most important contributions made by this study. As a result of this inquiry, the amount of processing time involved in the video processing module has been demonstrated. The chunk size is read by the FFMpeg utility that comes from the sender. The chunk is then divided into the nearest full frame, which ultimately results in a chunk size dictated by the user's input.

Figure 4 depicts the time required by the video processing module to divide the video file into many chunks. This time is displayed in the form of a delay. The results demonstrate

a straightforward and easy-to-understand correlation between the file size and the latency progression. According to the findings, the method does not result in an unanticipated increase in the required time. However, the generation of chunks is partly responsible for the overall delay. However, this is necessary to achieve a high level of security.

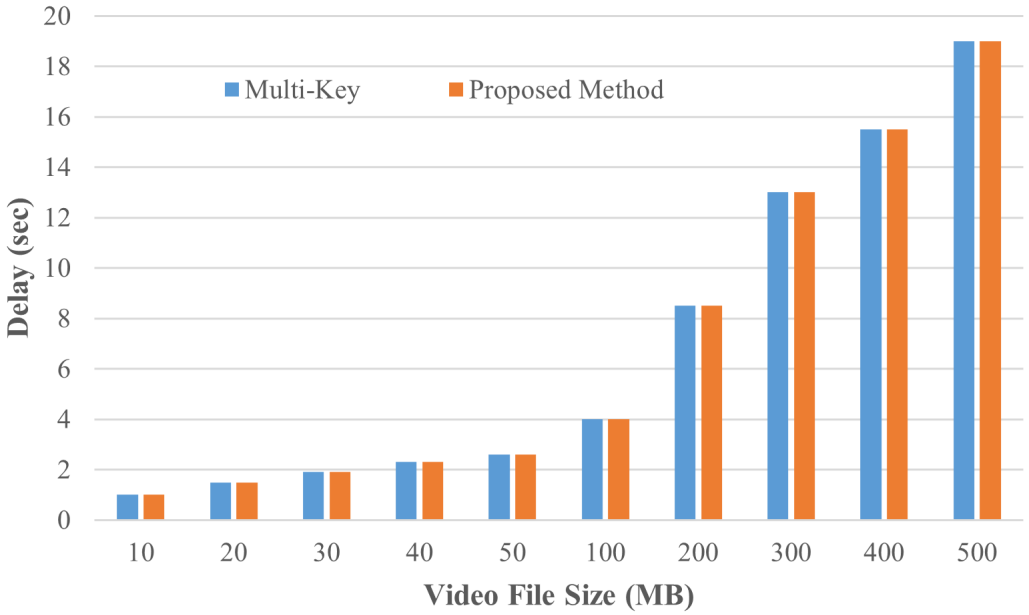

Figure 4. The length of time it takes the video processing module to divide the video into chunks (Compared with Multi-Key [13]).

4.2. Time to Generate the Keys

In this experiment, the amount of time needed to produce the key was measured to draw an assessment regarding the multi-key's effect on the encryption and decryption process. When it comes to encryption and decryption, the process and equation set that is utilised to derive the keys are exactly the same. Hence, in order to facilitate analysis, the delay that was computed for the encryption process is implemented in this section.

This section compares the key generation technique proposed in this work with the method proposed in [13]. The proposed method uses the equation set for dynamic key generation, and in [13], only the ECC equation is used.

The amount of time necessary to generate each key is illustrated in Figure 5. The public key of the receiver, in addition to some of the video data, is utilised as the foundation for the key that will be used to encrypt the first video chunk. The receiver details, the receiver's public key, and the previously computed key are used to acquire the remaining keys. Because the duration of the parameters remains the same throughout this process, the amount of time that passes between each key is not subject to much variation. When contrasted with the multi-key approach described in [13], the findings that were obtained are discussed. As a result of the utilised equation set, the delay consumed by the proposed approach is slightly increased. The key for each video chunk is generated with the help of an equation that was chosen at random.

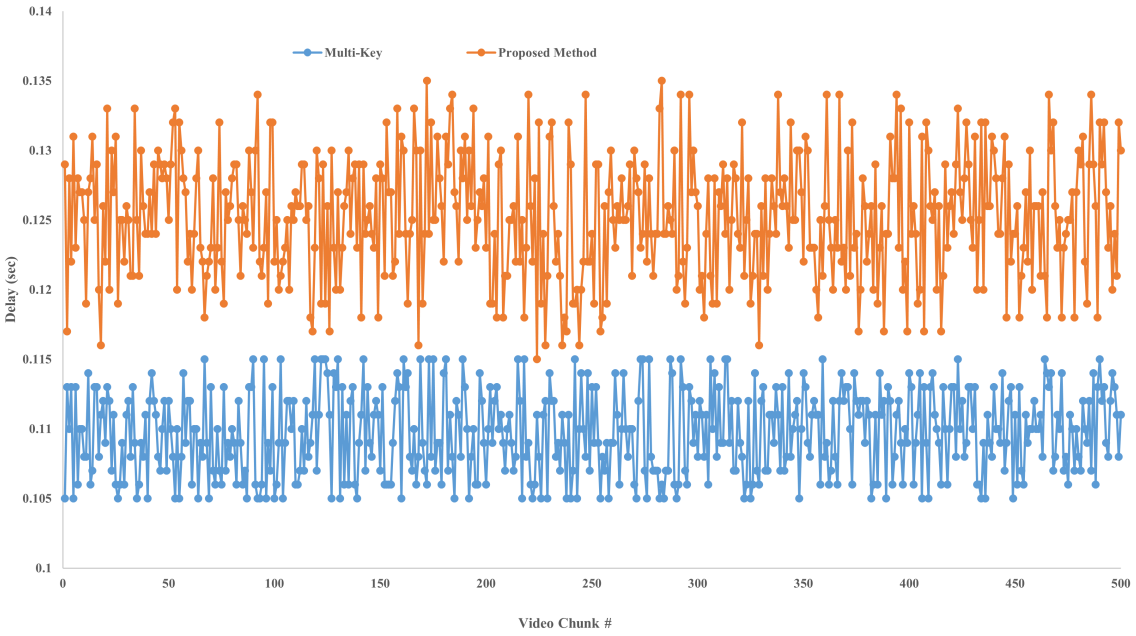

Figure 5. Multiple key generation delay (Compared with Multi-Key [13]).

4.3. Time to Encrypt the Video Chunks

In this section, we discussed the length of time required to encrypt each chunk. The implementation assumes that each chunk's size is 1 MB, despite the fact that the video processing modules divide the video file into chunks for each full frame. Each chunk is encrypted using the algorithm detailed in Algorithm 1. As mentioned above, the key size of 128 bits is used by the AES to encrypt the video chunks.

The amount of delay that is involved in the chunk encryption process is illustrated in Figure 6. The video chunks are encrypted using the AES module before being sent out into the network for streaming. The chunks are virtually identical in that their size ranges from 1 megabyte (MB) to 1.2 megabytes (MB), and the amount of time required to encrypt each chunk similarly ranges from 0.9 s to 1.2 s in the multi-key [13] method. In the proposed method, the encryption time ranges from 1.5 s to 2 s. The increase in the delay is observed due to the equation set and random selection of the equation for generating the key for each chunk.

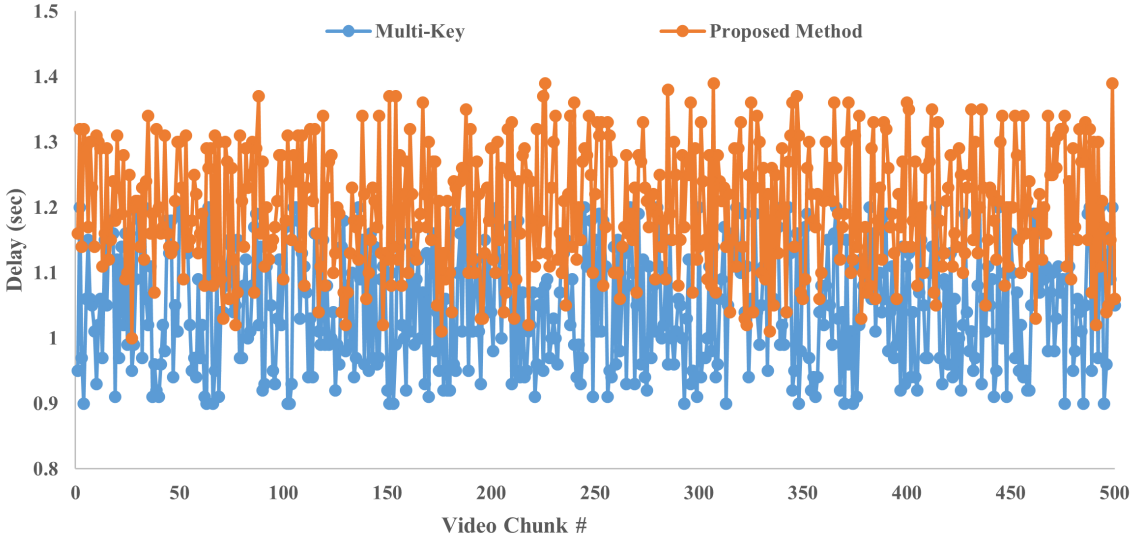

Figure 6. Time consumed to encrypt the video chunks (Compared with Multi-Key [13]).

4.4. Time to Encrypt the Video Files

This section gives information regarding the total amount of time that is necessary to finish the execution of the pipeline, which begins with the creation of video chunks and ends with encrypted video chunks. This section begins with creating video chunks and ends with encrypted video chunks. During the course of our investigation, we tested out a number of different sizes for video files, including 1, 2, 5, 10, 20, 30, 40, 50, 100, 200, 300, 400, and 500 megabytes. These file sizes have been selected with the applications that support real-time streaming in consideration. In addition to this, one of our goals was to investigate the behaviour of the increasing delay.

The latency experienced by the entire procedure is illustrated in Figure 7. According to the data, the process's duration increases linearly as the size of the file grows. This is normal for all applications. In addition, it is important to note that the additional delay caused by every segment of the video clip is depicted in the figure. On the receiver's end, neither the video streaming nor the playback is being disrupted in any way. The results demonstrate that the increased complexity using the equation set in the proposed method does not increase the delay much.

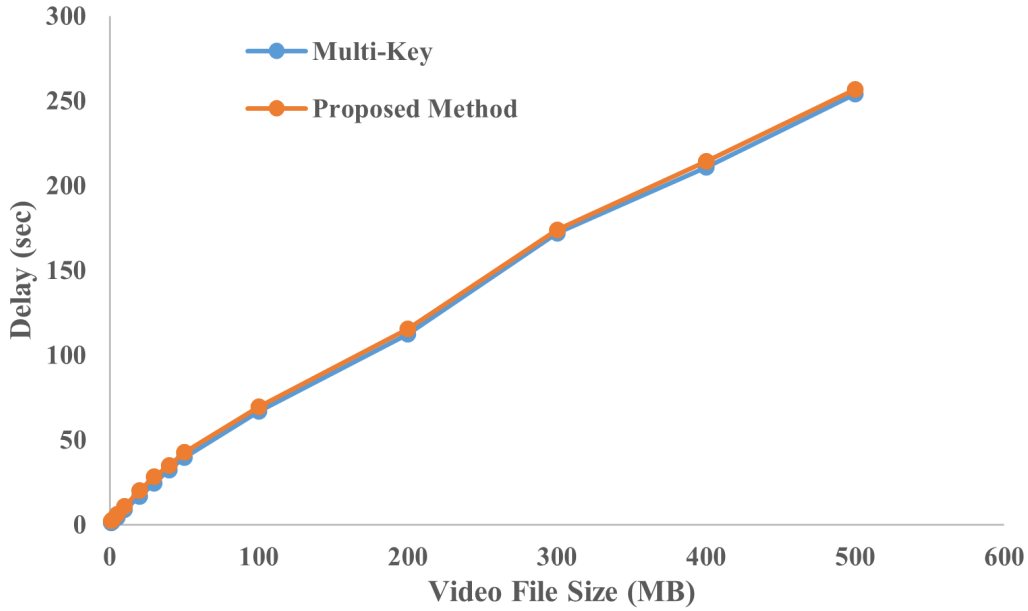

Figure 7. End-to-end encryption delay (Compared with Multi-Key [13]).

4.5. Number of Keys Generated

The number of generated keys is proportional to the number of generated video segments. This statistic was considered for the research because the proposed solution incorporates multiple key technologies for enhanced security. Multiple keys do not affect memory usage because the transmitter and receiver only temporarily store the keys. In addition, since each key is used only once, an increase in the number of keys has no effect on the retrieval delay.

4.6. End-to-End Decryption Delay

This section has looked at how long it takes for the receiver-side pipeline to process data. The receiver application receives the video chunks that have been encrypted, and it decrypts them in the order that they were received. Later, the chunks are assembled and displayed on a certain device. In this section, the length of time required for this processing cycle has been taken into consideration. Due to the fact that the decryption process is affected by the transmission media, receiver applications are required to wait until they have received the complete chunk before beginning to process it. When compared

to encryption, the latency could potentially become even longer. The time taken by the proposed method is compared with the method discussed in [13] and the results are shown in Figure 8.

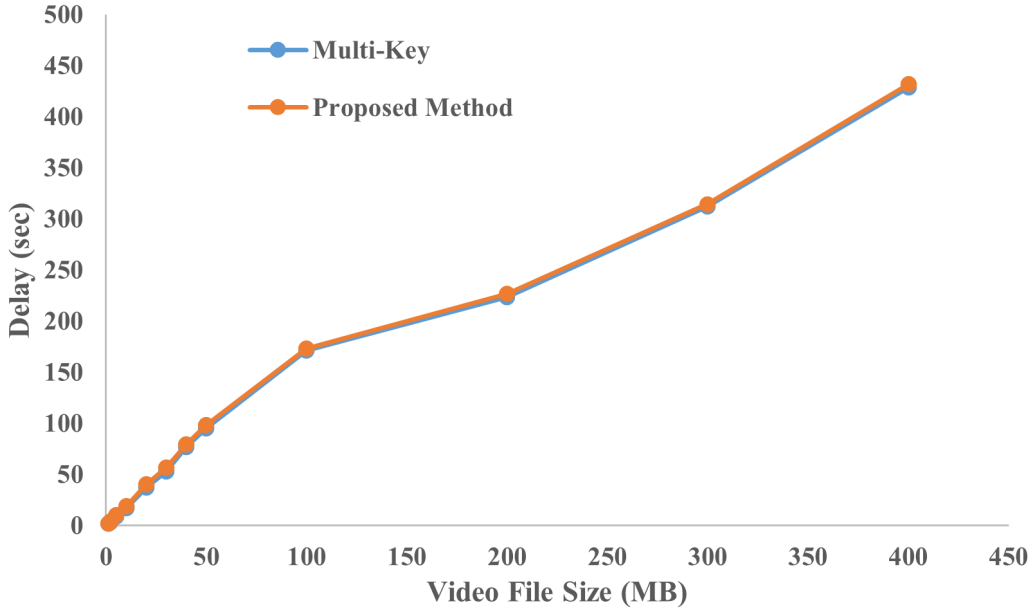

Figure 8. End-to-end delay in decryption (Compared with Multi-Key [13]).

5. Conclusions

This study presents a novel and secure system architecture. The platform will encrypt and decrypt the video file using the suggested cryptographic method. In this study, we present a hybrid and multi-key cryptography technique. It encrypts the video data using RSA, equation set, and AES.

The proposed multi-key approach divides the video into numerous pieces, each of which is encrypted using a different key. The recipients are not aware of these keys. The key is generated by the receiving program using the received encrypted data. Since the proposed solution is dynamic and automatic, the receiver program can begin decryption once it receives the video chunks.

The Android version of the app was developed on the receiving end. The proposed strategy and testing were implemented in Java-based server-side software. Individual modules were created, each tailored to the tasks considered by the proposed approach. The program was then put through its paces using video files of varying lengths. The results confirmed the app's continuous delay and demonstrated its ability to facilitate real-time video conversation. The equation set, RSA, and AES were all used in the hybrid method. These cryptographic methods make video material more secure.

The suggested platform encrypts and streams across the network while securing the video contents using a dynamic key, all while utilising the recipient's credentials and device details, making it ideal for use in video-on-demand applications. All created keys and encrypted content can be stored locally on the target device due to configurable program settings. This technology is very helpful for copyright protection because it prevents the data from being transmitted. More individuals are signing up for paid video services, which is good news for the industry's bottom line. The ultimate long-term objective is to create a state-of-the-art, thoroughly implemented security system for live-streamed video via the internet.

Author Contributions: Conceptualisation, Y.F.; formal analysis, Y.F. and A.L.; project administration, R.M.; Software, Y.F. and R.M.; supervision, A.L.; writing—original draft, R.M.; writing—review and editing, R.M., Y.F., and A.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. OTT Video-Worldwide. Available online: https://www.statista.com/outlook/amo/media/tv-video/ott-video/worldwide (accessed on 17 November 2023).
2. Stallings, W. *Network Security Essentials Applications and Standards*, 5th ed.; Pearson Education: London, UK, 2013.
3. Murtaza, A.; Hussain Pirzada, S.J.; Jianwei, L. A New Symmetric Key Encryption Algorithm With Higher Performance. In Proceedings of the 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), Sukkur, Pakistan, 30–31 January 2019; pp. 1–7. [CrossRef]
4. Kansal, S.; Mittal, M. Performance evaluation of various symmetric encryption algorithms. In Proceedings of the 2014 International Conference on Parallel, Distributed and Grid Computing, Solan, India, 11–13 December 2014; pp. 105–109. [CrossRef]
5. Kumar, S.; Gaur, M.S.; Sagar Sharma, P.; Munjal, D. A Novel Approach of Symmetric Key Cryptography. In Proceedings of the 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), London, UK, 28–30 April 2021; pp. 593–598. [CrossRef]
6. *NIST.FIPS.197*; Advanced Encryption Standard (AES). National Institute of Standards and Technology: Gaithersburg, MD, USA, 2001. [CrossRef]
7. Shen, Y.; Sun, Z.; Zhou, T. Survey on Asymmetric Cryptography Algorithms. In Proceedings of the 2021 International Conference on Electronic Information Engineering and Computer Science (EIECS), Changchun, China, 23–26 September 2021; pp. 464–469. [CrossRef]
8. Kumar, S.; Singh, B.K.; Akshita; Pundir, S.; Batra, S.; Joshi, R. A survey on Symmetric and Asymmetric Key based Image Encryption. In Proceedings of the 2nd International Conference on Data, Engineering and Applications (IDEA), Changchun, China, 23–26 September 2020; pp. 1–5. [CrossRef]
9. Rivest, R.L.; Shamir, A.; Adleman, L. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Commun. ACM* **1978**, *21*, 120–126. [CrossRef]
10. Koblitz, N. Elliptic Curve Cryptosystems. *Math. Comput.* **1987**, *48*, 203–209. [CrossRef]
11. Menezes, A.J.; Vanstone, S.A. Elliptic curve cryptosystems and their implementation. *J. Cryptol.* **1993**, *6*, 209–224. [CrossRef]
12. Koblitz, N.; Menezes, A.; Vanstone, S. The State of Elliptic Curve Cryptography. *Des. Codes Cryptogr.* **2000**, *19*, 173–193. [CrossRef]
13. Fouzar, Y.; Lakhssassi, A.; Ramakrishna, M. A Novel Hybrid Multikey Cryptography Technique for Video Communication. *IEEE Access* **2023**, *11*, 15693–15700. [CrossRef]
14. Chia, J.; Heng, S.H.; Chin, J.J.; Tan, S.Y.; Yau, W.C. An Implementation Suite for a Hybrid Public Key Infrastructure. *Symmetry* **2021**, *13*, 1535. [CrossRef]
15. Obaidat, M.; Brown, J.; Obeidat, S.; Rawashdeh, M. A Hybrid Dynamic Encryption Scheme for Multi-Factor Verification: A Novel Paradigm for Remote Authentication. *Sensors* **2020**, *20*, 4212. [CrossRef] [PubMed]
16. Saeed, Z.R.; Zakiah Ayop, N.; Baharon, M. Improved cloud storage security of using three layers cryptography algorithms. *Int. J. Comput. Sci. Inf. Secur.* **2018**, *16*, 35–39.
17. Al-Dhuraibi, Y.; Paraiso, F.; Djarallah, N.; Merle, P. Elasticity in cloud computing: State of the art and research challenges. *IEEE Trans. Serv. Comput.* **2017**, *11*, 430–447. [CrossRef]
18. Hodowu, D.K.M.; Korda, D.R.; Ansong, E.D. An enhancement of data security in cloud computing with an implementation of a two-level cryptographic technique, using AES and ECC algorithm. *Int. J. Eng. Res. Technol.* **2020**, *9*, 639–650.
19. Lee, B.H.; Dewi, E.K.; Wajdi, M.F. Data security in cloud computing using AES under HEROKU cloud. In Proceedings of the 2018 27th Wireless and Optical Communication Conference (WOCC), Hualien, Taiwan, 30 April–1 May 2018; pp. 1–5.
20. Zhu, Y.; Fu, A.; Yu, S.; Yu, Y.; Li, S.; Chen, Z. New algorithm for secure outsourcing of modular exponentiation with optimal checkability based on single untrusted server. In Proceedings of the 2018 IEEE International Conference on Communications (ICC), Kansas City, MO, USA, 20–24 May 2018; pp. 1–6.
21. Micciancio, D.; Tessaro, S. An equational approach to secure multi-party computation. In Proceedings of the 4th Conference on Innovations in Theoretical Computer Science, Berkeley, CA, USA, 9–12 January 2013; pp. 355–372.
22. Giacon, F.; Kiltz, E.; Poettering, B. Hybrid encryption in a multi-user setting, revisited. In *Public-Key Cryptography–PKC 2018, Proceedings of the 21st IACR International Conference on Practice and Theory of Public-Key Cryptography, Rio de Janeiro, Brazil, 25–29 March 2018*; Proceedings, Part I 21; Springer: Berlin/Heidelberg, Germany, 2018; pp. 159–189.
23. Chaudhari, N.; Saini, M.; Kumar, A.; Priya, G. A Review on Attribute Based Encryption. In Proceedings of the 2016 8th International Conference on Computational Intelligence and Communication Networks (CICN), Tehri, India, 23–25 December 2016; pp. 380–385. [CrossRef]

24. Niu, S.; Chen, L.; Liu, W. Attribute-Based Keyword Search Encryption Scheme with Verifiable Ciphertext via Blockchains. In Proceedings of the 2020 IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 11–13 December 2020; Volume 9, pp. 849–853. [CrossRef]
25. Wang, C.; Luo, J. An Efficient Key-Policy Attribute-Based Encryption Scheme with Constant Ciphertext Length. *Math. Probl. Eng.* **2013**, *2013*, 810969. [CrossRef]
26. Hohenberger, S.; Waters, B. Attribute-based encryption with fast decryption. In *Public-Key Cryptography–PKC 2013, Proceedings of the 16th International Conference on Practice and Theory in Public-Key Cryptography, Nara, Japan, 26 February–1 March 2013*; Proceedings 16; Springer: Berlin/Heidelberg, Germany, 2013; pp. 162–179.
27. Boldyreva, A.; Imai, H.; Kobara, K. How to Strengthen the Security of RSA-OAEP. *IEEE Trans. Inf. Theory* **2010**, *56*, 5876–5886. [CrossRef]