IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 11 January 2023, accepted 30 January 2023, date of publication 6 February 2023, date of current version 21 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3242616

RESEARCH ARTICLE

A Novel Hybrid Multikey Cryptography Technique for Video Communication

YOUCEF FOUZAR[1], AHMED LAKHSSASSI[1], (Senior Member, IEEE), AND M. RAMAKRISHNA[2], (Member, IEEE)

[1]Department of Computer Science and Engineering, Université du Québec en Outaouais, Gatineau, QC J8X 3X7, Canada
[2]Department of Data Science and Computer Applications, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, India

Corresponding authors: Youcef Fouzar (fouy03@uqo.ca) and M. Ramakrishna (ramakrishna.m@manipal.edu)

ABSTRACT Online video streaming is becoming more widespread in people's everyday entertainment routines. Protecting copyright and piracy has become a key concern in real-time video streaming systems. This research provides a revolutionary multi-key and hybrid cryptography approach to offer security. This work describes the software implementation of video encryption and decryption employing continuous systems based on the Elliptic Curve Cryptography approach as pseudorandom encryption key generators. This approach creates several keys to encrypt and decode small chunks of video files that are produced dynamically based on the video data. The suggested approach was implemented on the Android platform, where applications for sender and recipients had been created to enable streaming. The security and performance of the proposed system have been examined by implementing it on devices and streaming videos. The outcomes demonstrate superiority in terms of performance and security.

INDEX TERMS Asymmetric key cryptography, multi-key encryption.

I. INTRODUCTION

Real-time media streaming has become a commodity due to rapid advancements and developments in our Internet infrastructure and applications that drive these technologies [1], [2]. Streaming media is the continuous delivery of media data, such as audio or video, over the Internet, where the content is presented to the end-user before it has been completely downloaded. Due to the growing popularity of video conferencing, web-based television services, e-learning, telemedicine, and popular Internet-based businesses like YouTube and Netflix offer live media streaming services to their corporate and individual users [3], [4]. As a result, there is more sharing of Internet traffic. Additionally, the Internet is a decentralized network; therefore, anyone can connect from anywhere and share any media data [5].

The increase in the video traffics of OTT (Over-The-Top) services [6], [7], and similar applications have shown significant concern for security and privacy. Both consumers and producers have been experiencing illegal sharing of media content and pirated videos of sensitive data. These disrupt mainly the data related to telemedicine and real-time video streaming.

The following security measures are used in content protection:

- forensic watermarking to prevent content re-acquisition during rendering;
- trusted compute environment to prevent access during decoding; and
- encryption to prohibit access to the content during transit.

New content protection strategies that rely less on hardware are required with the introduction of next-generation video and the rising popularity of embedded devices for content consumption.

Hence, a suitable cryptography technique is needed to handle the issues of video communication. Here, the most critical challenges that need to be focused on are authentication, encryption, and key management [8], [9].

Many cryptography techniques are implemented in the applications to improve media data security. Symmetric and asymmetric key cryptography is the techniques that are available to achieve security in communication. Symmetric key

The associate editor coordinating the review of this manuscript and approving it for publication was Chien-Ming Chen.

cryptography is the one most used in multimedia communication [10], [11], [12]. Advanced Encryption Standard (AES) [13] is the most efficient and commonly used symmetric encryption. Websites and web browsers use 128-bit AES to provide security over Internet communication. The key management method has been problematic in this procedure due to the implementation of the Secure Real-time Transport Protocol (SRTP). As the network is decentralized, key management becomes a challenge. In many techniques, the key and algorithms cannot be split effectively to improve the security of the Internet. Hence, many research works are going on to realize asymmetric key cryptography techniques [14], [15]. Asymmetric cryptography methods such as Rivest-Shamir-Adleman (RSA) [16] and Elliptic Curve Cryptography (ECC) [17], [18], [19] have been explored and gaining popularity to overcome the challenges of symmetric cryptography.

In a video streaming application, the video data is divided into multiple chunks and then streamed using streaming protocols. The majority of the traditional encryption and decryption methods use symmetric key cryptography, but the key exchange methods in these techniques lead to security bleaches. Hence, asymmetric key cryptography techniques help enable higher security for video content streamed over the Internet. Figure 1 shows the asymmetric key cryptography techniques in video communication. The keys used in this method are static; hence, a dynamic key generation method is needed. This paper has developed a novel method that uses asymmetric key cryptography to encrypt video chunks. This work involves designing and developing a novel equation-based multi-key encryption technique and video attribute-based decryption key generation method.

This works aims at developing a multi-key cryptography technique for video streaming applications. The features of this method are as follows:

- RSA and ECC-Based Method: Both methods provide higher security in video communication. The RSA is used for encrypting the Video ID, and ECC is for generating encrypted video chunks.
- Multi-key Technique: The separate keys are generated for each chunk of video data, and a separate key for video metadata.
- Automatic Key Generation: An equation-based key generation method implemented to achieve dynamic and automatic key generation. This feature enables the algorithm to generate a unique key for each video stream.

The rest of the paper is organized as follows: in Section II, we discuss the related works. Section III presents the proposed mechanism: the evaluation and numerical results of the algorithm detailed in Section IV. Finally, the conclusions drawn are described in Section V.

II. RELATED WORK

Zia et al. [20] proposed a pseudorandom number generator-based chaos theory capable of generating a unique and independent number that can be used in cryptography techniques. This method helps in automatic and adaptive random number generation. Kezia and Sudha [21] developed a novel video encryption scheme based on chaotic maps. Here, the encrypted video sequence is taken, and then it is split up into frames. The frames are broken into macro-blocks for the operation when frames are large. The high-dimensional Lorenz chaotic system is employed to confuse the position of the pixels, and the multi-key concept is used to improve the security of the cryptosystem against attacks.

Khan et al. [22] proposed an ECC-based authentication and encryption technique for IoT applications. In this work, the computational cost and delay in processing the medical sensed data have been analyzed and demonstrated the fast processing of ECC. Imam et al. [23] reviewed RSA-based cryptographic techniques and suggested the suitability of the crypto techniques for various applications.

Sen et al. [24] studied the performance of ECC-based cryptography techniques on video data. The ECC performs better than any other asymmetric crypto technique because of the smaller key and faster encryption and decryption operations. In [25] and [26], a hybrid crypto approach that uses RSA or AES with ECC for video encryption has been studied, and the performance of the techniques measured. Chen and Ye [27] implemented an image encryption method using hash SHA-3, RSA and compressive sensing. This hybrid model achieves higher security using chaotic sequence along with the latter listed methods.

Hegde and Jagadeesha [28] designed a crypto technique using ECC. The uniqueness of the method is that it uses multiple elliptic curves to improve the robustness of the data. The metadata are separated and encrypted and then embedded into a video using Optimized Modified Matrix Encoding.

Murad Rahouma [29] have studied two-tier and three-tier hybrid cryptography models applied for cloud security. s. Ghosh et al. [30] proposed a hybrid method to achieve confidentiality and security of the data on the Internet. Zhang and Gao [31] proposed a method for the video data encoded using a layered coding method. Yu et al. [32] have discussed the applications of the hybrid encryption algorithm in software securities. The work discusses the uses of hybrid methods in enhancing security in video surveillance software.

Khan et al. [33], to accomplish the Hybrid encryption technique, data encryption techniques using the Fibonacci series, XOR logic, PN sequence are studied, analyzed, and their performance is compared in this work. The message is divided into three parts, and these three different techniques are applied to these parts, and the performance is again analyzed. The basis of this work is the application of these three different methods to different parts of the same message along with two keys, namely, segmenting key and encrypting key, to provide further authentication and validation.

Chowdhary et al. [34] have proposed an analysis for performing image encryption and decryption by hybridization of Elliptic Curve Cryptography (ECC) with Hill Cipher (HC), ECC with Advanced Encryption Standard (AES) and ElGamal with Double Playfair Cipher (DPC). Dave and

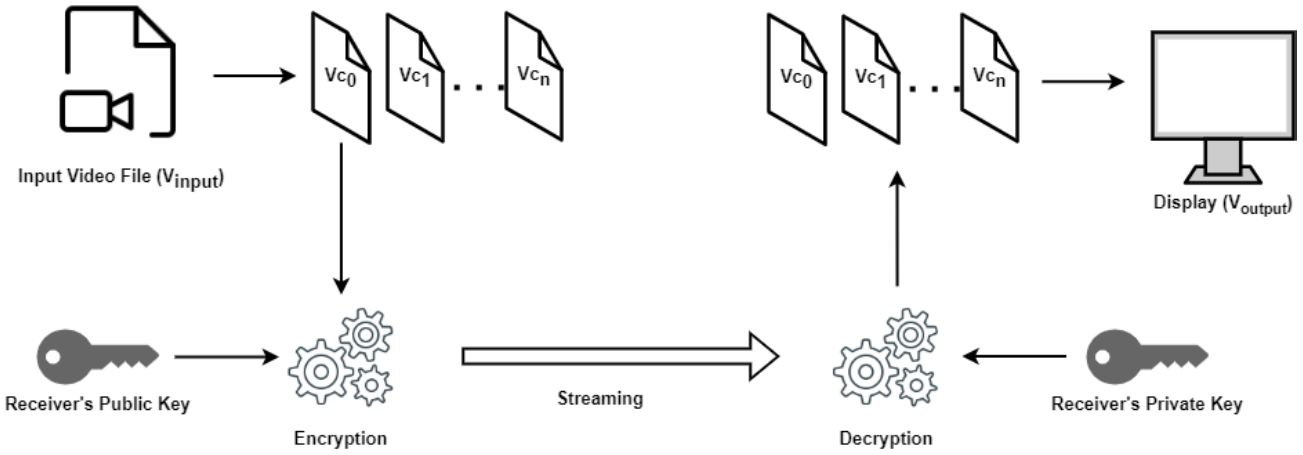


FIGURE 1. Secured video communication using asymmetric key cryptography.

Gayathri [35], the pros and cons of storing biometric data on the cloud are discussed. For unimodal biometric templates, the paper shows a hybrid cloud-based encryption method. A generic algorithm has been proposed. Additionally, the overall impact of the existing algorithms and the challenges associated with their use are briefly discussed.

Hamdi et al. [36] have proposed a hybrid encryption algorithm based on block and stream ciphers using chaotic systems. The proposed scheme adopts two primary operations one to generate a pseudorandom data block that will be used for stream cipher, and the second to create substitution and permutation tables in the initial step and perform rounds for confusion and diffusion processes in the block cipher.

Yu et al. [37] explored a unique hybrid model by leveraging the power of automation for security tests at the Video Acquisition/Aggregation level and amalgamating the best practices from traditional security tests done at the user Video Application level. This hybrid methodology covers all video streaming value chain phases, from origin to playback. Therefore, it achieves maximum test coverage across multiple playback devices under multiple workload conditions. They deploy real-world conditions in this methodology, including latency, delay, and concurrency.

Zhu et al. [38] proposed a hybrid encryption approach based on a blockchain. The processing delay of the encryption and decryption procedure is lengthened in this paradigm by the query time and security levels. Hosny et al. [39] developed a chaotic logistic map-based encryption method. To increase security, frame-level encryption has been used in this technology. Similarly, Alarifi et al. [40] proposed a cryptosystem combining DNA encoding sequences, Arnold chaotic map, and modified Mandelbrot sets. However, the approaches work well for applications that stream video on demand and do not anticipate real-time streaming.

From this literature, it is observed that the hybrid model is the suitable method to enhance the robustness of the secured data. Additionally, the ECC-based crypto techniques are more suitable for real-time video streaming applications as ECC generates a small key and is fast in processing. However, asymmetric methods increase the complexity of the decryption; hence hybrid model using RSA and ECC is considered in this work.

III. PROPOSED METHOD

In this work, we have proposed a novel multi-key cryptography technique to improve security in video communication. The proposed method uses RSA and ECC to achieve the asymmetric crypto technique.

A. PROBLEM STATEMENT

In a video streaming application, the video is streamed from the media server to the client devices on-demand basis. Securing digital video content involves the following: conditional access, user authentication, content copy control, and video content tracking. These security measures are generally realized using cryptography techniques. However, achieving a complete solution for digital video security is a research challenge.

Much research has been carried out in cryptography to explore the advantages of asymmetric key cryptography methods to overcome key management challenges. The existing methods do not support dynamic and automatic multi-key techniques to enable higher security in video communication applications. As a result, an automatic and dynamic key management method is needed. Hence, this work focuses on a multi-key encryption technique based on RSA and ECC.

B. PROPOSED ALGORITHMS

The proposed method aims to generate multiple dynamic keys based on RSA and ECC asymmetric key cryptography techniques. The goal of the proposed techniques are:

- secure the video data based on the content and receiver's unique identifications

- to take the advantage of the hybrid crypto techniques, the proposed uses RSA, ECC and AES techniques
- improving the security by multi-key encryption technique
- increasing the security with the video chunks
- reduce multi-key management using automatic key generation methods

The goals are achieved using the proposed key generations model shown in Figure 2.

The process is initiated by passing video data to the key generation module. The module requires the receiver's Public Key R_{Pkey} and MAC address R_{mac}; these improve the uniqueness of the keys generated in the process. These attributes are required throughout video communication; hence, they are stored on the sender's side. Initially, the video is divided into multiple chunks of size 1 MB. These video chunks are used individually in the key generation.

Algorithm 1 and Figure 3 discuss the steps involved in the key generation. The encryption of video chucks starts with creating the unique video identification V_{ID}. The V_{ID} is generated using the first video chunk Vc_0. In this step, the module fetches the first 16 bytes from the Vc_0 before the encryption. Then, it converts it into a base64 string format. Later, it is used as V_{ID} in the key generation; therefore, the V_{ID} is stored temporarily in a file for quick access. The first video chuck is encrypted using the V_{ID} and RSA crypto technique.

Algorithm 1 Encryption Flow

1: Input video file V_{input}
2: Generate video chunks Vc_i from V_{input}
3: Fetch the receiver's public key of the R_{Pkey}
4: Collect receiver's MAC address R_{mac}
5: Generate V_{ID} using Vc_0
6: Store V_{ID} in a temporary file
7: Encrypt Vc_0 using *RSA*
8: Generate $Key_a \leftarrow x^3 + V_{ID} * x + R_{mac}$
9: Encrypt Vc_1 using Key_a and AES
10: **for** i:=2 **do**n
11: Generate $Key_a \leftarrow x^3 + Key_a * x + R_{mac}$
12: Encrypt Vc_i using Key_a and AES
13: **end for**

The key for each video chuck is created as follows: the V_{ID} generates the subsequent key. Here, the method uses Public Key R_{Pkey} and MAC address R_{mac} of the receiver to generate the key for the second video chuck. The method derives the key from the ECC equation, i.e.,

$$y^2 = x^3 + ax + b \tag{1}$$

The proposed method considers V_{ID} as a and R_{mac} as b in the ECC equation (Eq.1). Hence, the modified equation is as follows:

$$Key_a = x^3 + V_{ID} * x + R_{mac} \tag{2}$$

Eq. 2 is used only for the first video chuck. The video chuck uses that is from the second chuck; it uses Eq. 3. Here, the algorithm considers previously computed Key_a instead of V_{ID}. This method is continuous for all the remaining chucks in the video.

$$Key_a = x^3 + Key_a * x + R_{mac} \tag{3}$$

The unique key that is generated for each video chuck is then used for encrypting the video chuck using the AES algorithm. The proposed method does not share any keys in this multi-key and hybrid method. As described in Algorithm 1, the key is generated on the fly for video communication; hence maximum security can be achieved. Another essential feature of the proposed method is that even if one chuck is compromised using brute force methods, the rest of the video data is secure.

IV. EXPERIMENTATION

The mobile platform is used to implement the suggested cryptography technique. Here, an android-based application is used to implement the encryption and decryption processes and video streaming.

For streaming, the sender devices store the video content and video metadata. When a receiver requests to stream a video file, the sender application starts. Before sending the video content to the recipient, the sender, an authorized video distributor, verifies the recipient's identity. A database is used to maintain receiver information. Information about the recipient is gathered during the sign-up process. All information, including the username, password, public key, and device information like the MAC address, is retrieved and stored in the sender's database.

The main tasks at the receiver are decryption and video playback. The module receives encrypted video chunks and decrypts them using the proposed module. The necessary keys are obtained from the receiver's database. As a result, no key exchange occurs in this method. The decryption employs the sender's public and receiver's private keys. To obtain the first video parts, the module employs an RSA implementation. The video data is then displayed on the device's display unit via the implemented application. This eliminates the need to save received video data.

Implementing a database is necessary for the applications' need to store the video metadata and receiver information. The following information is kept in the database on the sender side: Information about the recipient, including username, MAC address of the device, public key, and other account-specific information. The sender's public key and the communication session's video metadata are similarly stored in the receiver side database.

The dynamic key is automatically generated by the proposed approach using the ECC equation. The ECC is known for the trap-door mechanism; hence, the proposed module uses it. Along with the previously generated key that was covered in the previous section III, the module also takes into account the receiver's public key and MAC address.

The sender-side module reads the video file, creating the chunks. The video chunks in this instance were generated

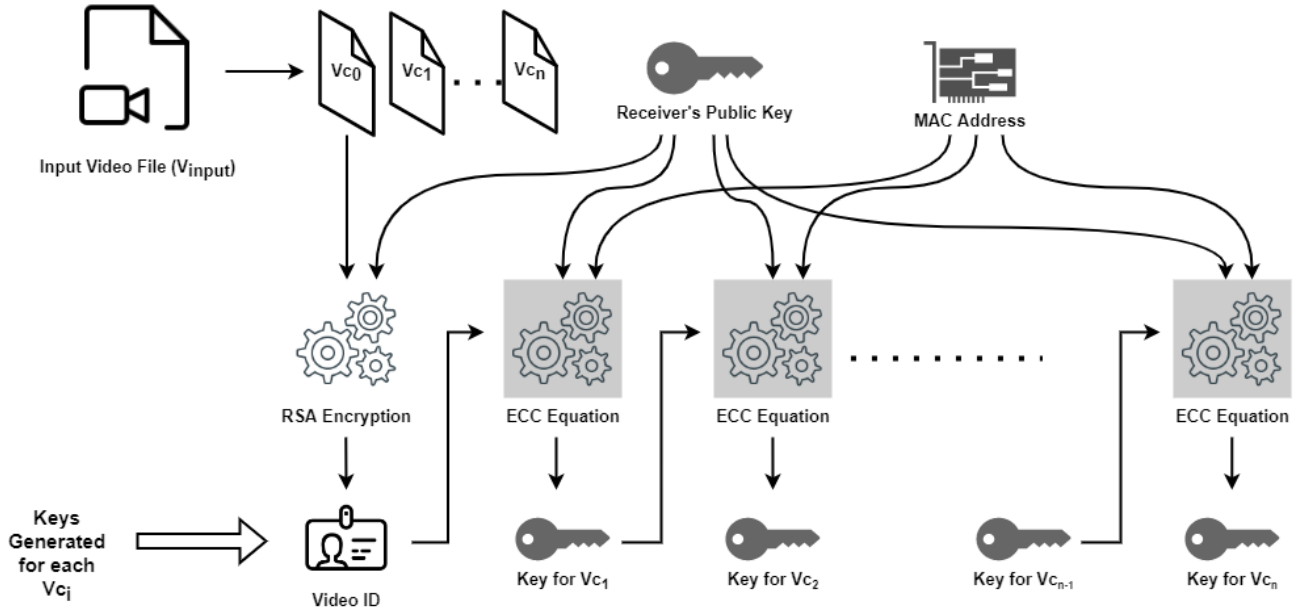


FIGURE 2. Block diagram of proposed key generation technique.

using the FFMpeg module. A unit size has been used to divide the video. The module generates the chunks to the nearest full frame, regardless of the chunk size, which can be any size. The proposed cryptography technique was evaluated in this section using parameters such as time to generate each key, time to encrypt files with varying file sizes, time to encrypt files with varying chunk sizes, Number of keys generated, and End to end processing delay from key generation to complete encryption. The same is held for decryption.

A. DELAY FOR SPLITTING FILE TO CHUNKS

The time it took the application to create the video chunks from the video file was measured using this metric. One of the key contributions of this work is the use of video chunk-based encryption. This investigation, therefore, demonstrates the processing time involved in the video processing module. The FFMpeg utility from the sender reads the chunk size. The chunk is then divided into the nearest full frame, resulting in a chunk size that is determined by user input.

The time needed by the video processing module to split the video file into several chunks is depicted in Figure 4. The outcome shows that the latency increases gradually as file size increases, which is simple to understand. The findings demonstrate that the technique does not unexpectedly lengthen the time. However, the delay as a whole is a result of chunk formation. To accomplish high security, however, this is necessary.

B. TIME TO GENERATE THE KEYS

To determine the impact of the multi-key in the encryption and decryption process, the time required to generate the key was calculated in this experiment. The method and equation used to derive the keys are the same for both encryption and decryption. So, for the sake of analysis, the delay calculated for encryption has been employed in this part.

Figure 5 depicts the time required to generate each key. The receiver's public key and partial video data serve as the basis for the key used to encrypt the first video chunk. The remaining keys are obtained using the receiver's MAC address, public key, and previously computed key. Since the length of the parameters is consistent during this procedure, the time between each key does not vary much.

C. TIME TO ENCRYPT THE VIDEO CHUNKS

This section has talked about how long it takes to encrypt each chunk. Although the video processing modules split the video file into chunks for each full frame, the implementation assumes the chunk size to be 1 MB.

Figure 6 depicts the delay involved in the check encryption. The video chunks are fed into the AES module before being transmitted. The chunks are mostly the same in that it varies between 1MB and 1.2MB, and the time taken to encrypt each also varies between 0.9 sec to 1.2 sec.

D. TIME TO ENCRYPT THE VIDEO FILES

This section details the execution time for the pipeline, which starts with the production of video chunks and ends with encrypted video chunks.

The process's end-to-end latency is depicted in Figure 7. The findings indicate that as the file size increases, the time increases progressively. This is typical for all applications. Furthermore, it is significant to note that Figure 7 depicts the added delay by each chunk of the video clip. The video streaming and playback on the receiver side are unaffected.

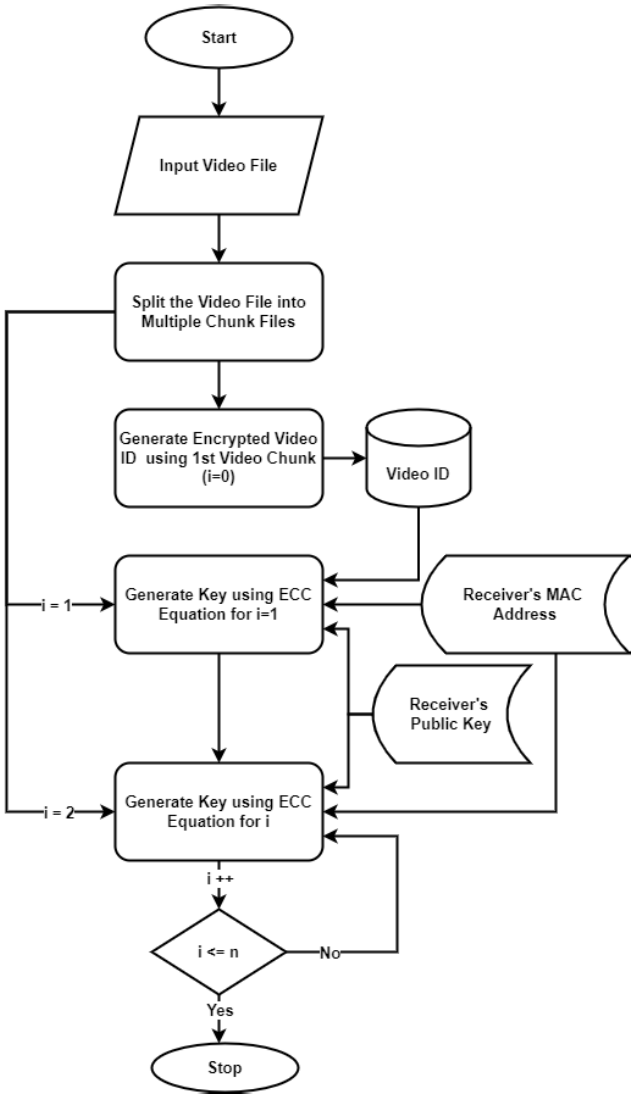


FIGURE 3. Block diagram of proposed key generation technique.

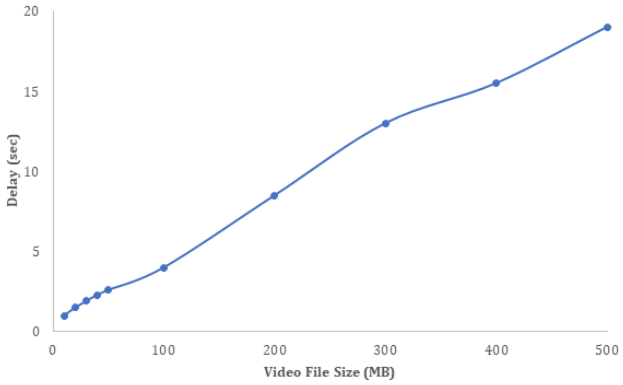


FIGURE 4. Time taken by the video processing module to split the video into chunks.

E. NUMBER OF KEYS GENERATED

The number of keys generated depends on the quantity of generated video chunks. This statistic has been considered for the study because the suggested solution uses multiple key technologies to provide improved security. Multiple keys

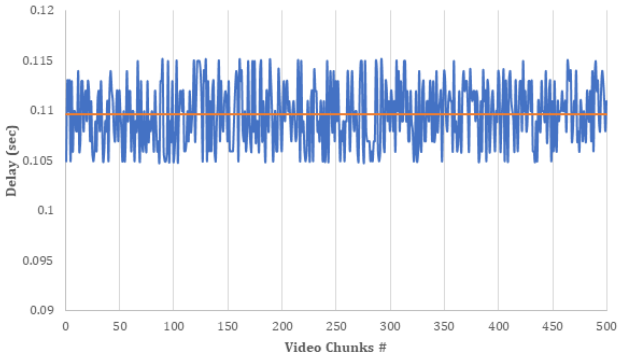


FIGURE 5. Delay in generating multiple keys.

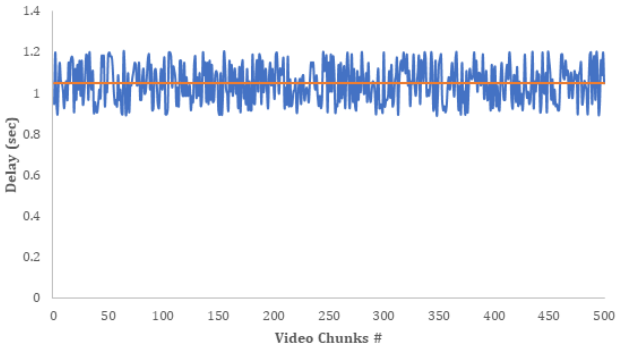


FIGURE 6. Time taken to encrypt the video chunks.

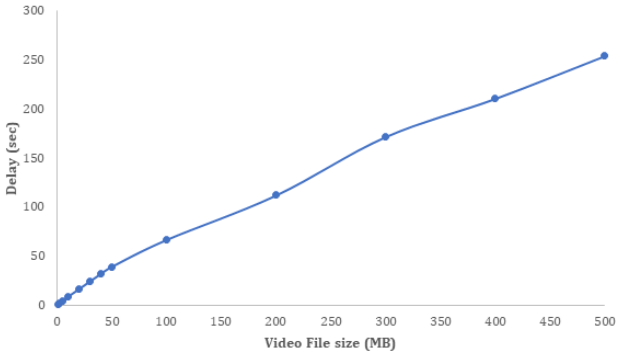


FIGURE 7. End-to-end encryption delay.

have no impact on memory use because the keys are only momentarily saved at the transmitter and receiver sides. Furthermore, because each key is only utilized once, an increase in the number of keys has no impact on the fetching delay.

F. END-TO-END DECRYPTION DELAY

In this section, the receiver side pipeline's processing time has been examined. The encrypted video chunks are delivered to the receiver application, which decrypts them sequentially. The chunks are then combined and displayed on a device. Here, it has been considered how much time this processing cycle takes. Because the transmission medium impacts the decrypting process, receiver apps must wait until they have

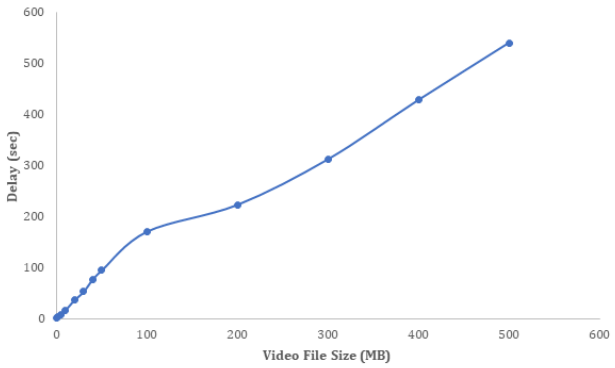


FIGURE 8. End-to-end decryption delay.

received the entire chunk before processing it. Comparing the delay to encryption might make it longer.

V. CONCLUSION

An innovative and secure platform has been presented in this work. The platform uses the suggested cryptographic method to encrypt and decrypt the video file. A hybrid and multi-key cryptography method has developed in this work. It secures the video contents using RSA, the ECC equation, and AES.

The multi-key solution that has been presented separates the video into many parts and then encrypts each chunk with a different key. The receivers do not have access to these keys. The receiver application generates the key using the encrypted chunks it has received. The receiver application starts the decryption process as soon as the video chunks are received because the suggested method is dynamic and automatic.

The application was created on the receiver side for the Android platform. A java-based server-side program has been created to implement the suggested method and testing. Based on the tasks taken into account in the suggested strategy, distinct modules have been developed. Then, video files of various sizes were used to test the application. The findings showed that the application's delay is consistent and supports real-time video communication. The hybrid approach uses the well-established AES, RSA, and ECC algorithms. These cryptography approaches improve the security of the video content.

The proposed platform is advantageous for video-on-demand applications since it encrypts and streams across the network while protecting the video contents using a dynamic key, all using the recipient's credentials and device information. Program settings can be made to temporarily store all generated keys and encrypted material locally on the destination device. By preventing the data from being transferred, this tool greatly aids in protecting copyright. This helps increase the number of people who subscribe to video services, which boosts earnings.

Future goals include expanding this work to multi-level security systems and creating a cutting-edge, deeply-implemented security system for online video streaming.

REFERENCES

[1] O. El Marai, T. Taleb, M. Menacer, and M. Koudil, "On improving video streaming efficiency, fairness, stability, and convergence time through client–server cooperation," *IEEE Trans. Broadcast.*, vol. 64, no. 1, pp. 11–25, Mar. 2018.

[2] Z. Lu and I. Nam, "Research on the influence of new media technology on internet short video content production under artificial intelligence background," *Complexity*, vol. 2021, pp. 1–14, Jan. 2021. [Online]. Available: https://ideas.repec.org/a/hin/complx/8875700.html

[3] F. Loh, F. Wamser, F. Poignée, S. Geißler, and T. Hoßfeld, "YouTube dataset on mobile streaming for internet traffic modeling, network management, and streaming analysis," *Sci. Data*, vol. 9, p. 293, Apr. 2022. [Online]. Available: https://figshare.com/articles/dataset/YouTube_Dataset_on_Mobile_Streaming_for_Internet_Traffic_Modeling_Network_Management_and_Streaming_Analysis/19096823

[4] D. Shamsimukhametov, M. Liubogoshchev, E. Khorov, and I. F. Akyildiz, "Youtube, netflix, web dataset for encrypted traffic classification," in *Proc. Int. Conf. Eng. Telecommun.*, 2021, pp. 1–5, doi: 10.21227/s7x7-wd58.

[5] *Cisco Annual Internet Report—Cisco Annual Internet Report (2018–2023) White Paper*. Accessed: Feb. 15, 2023. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[6] *Cisco Annual Internet Report—Cisco Annual Internet Report Highlights Tool*. Accessed: Feb. 15, 2023. [Online]. Available: https://www.cisco.com/c/en/us/solutions/executive-perspectives/annual-internet-report/air-highlights.html

[7] A. Rao, A. Legout, Y.-S. Lim, D. Towsley, C. Barakat, and W. Dabbous, "Network characteristics of video streaming traffic," in *Proc. 7th Conf. Emerg. Netw. Exp. Technol.*, Dec. 2011, pp. 1–12.

[8] X. Huang, D. Arnold, T. Fang, and J. Saniie, "A chaotic-based encryption/decryption system for secure video transmission," in *Proc. IEEE Int. Conf. Electro Inf. Technol. (EIT)*, May 2021, pp. 369–373.

[9] A. Massoudi, F. Lefebvre, C. De Vleeschouwer, B. Macq, and J.-J. Quisquater, "Overview on selective encryption of image and video: Challenges and perspectives," *EURASIP J. Inf. Secur.*, vol. 2008, pp. 1–18, Jan. 2008.

[10] A. Murtaza, S. J. H. Pirzada, and L. Jianwei, "A new symmetric key encryption algorithm with higher performance," in *Proc. 2nd Int. Conf. Comput., Math. Eng. Technol. (iCoMET)*, Jan. 2019, pp. 1–7.

[11] S. Kansal and M. Mittal, "Performance evaluation of various symmetric encryption algorithms," in *Proc. Int. Conf. Parallel, Distrib. Grid Comput.*, Dec. 2014, pp. 105–109.

[12] S. Kumar, M. S. Gaur, P. S. Sharma, and D. Munjal, "A novel approach of symmetric key cryptography," in *Proc. 2nd Int. Conf. Intell. Eng. Manag. (ICIEM)*, Apr. 2021, pp. 593–598.

[13] M. Dworkin, E. Barker, J. Nechvatal, J. Foti, L. Bassham, E. Roback, and J. Dray, "Advanced encryption standard (AES)," Federal Inf. Process. Standard, Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. 197, 2001. Accessed: Feb. 15, 2023, doi: 10.6028/NIST.FIPS.197.

[14] Y. Shen, Z. Sun, and T. Zhou, "Survey on asymmetric cryptography algorithms," in *Proc. Int. Conf. Electron. Inf. Eng. Comput. Sci. (EIECS)*, Sep. 2021, pp. 464–469.

[15] S. Kumar, B. K. Singh, S. Pundir, S. Batra, and R. Joshi, "A survey on symmetric and asymmetric key based image encryption," in *Proc. 2nd Int. Conf. Data, Eng. Appl. (IDEA)*, Feb. 2020, pp. 1–5.

[16] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978, doi: 10.1145/359340.359342.

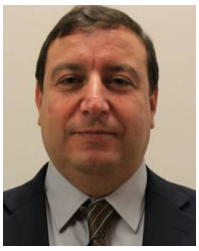
[17] N. Koblitz, "Elliptic curve cryptosystems," *Math. Comput.*, vol. 48, no. 177, pp. 203–209, 1987.

[18] A. J. Menezes and S. A. Vanstone, "Elliptic curve cryptosystems and their implementation," *J. Cryptol.*, vol. 6, no. 4, pp. 209–224, 1993. [Online]. Available: https://link.springer.com/article/10.1007/BF00203817

[19] N. Koblitz, A. Menezes, and S. Vanstone, "The state of elliptic curve cryptography," *Designs, Codes Cryptogr.*, vol. 19, nos. 2–3, pp. 173–193, Mar. 2000. [Online]. Available: https://link.springer.com/article/10.1023/A:1008354106356

[20] U. Zia, M. McCartney, B. Scotney, J. Martinez, and A. Sajjad, "A novel pseudo-random number generator for IoT based on a coupled map lattice system using the generalised symmetric map," *Social Netw. Appl. Sci.*, vol. 4, no. 2, pp. 1–17, Feb. 2022. [Online]. Available: https://link.springer.com/article/10.1007/s42452-021-04919-4

[21] H. Kezia and G. F. Sudha, "Encryption of digital video based on Lorenz chaotic system," in *Proc. 16th Int. Conf. Adv. Comput. Commun.*, Dec. 2008, pp. 40–45.
[22] M. A. Khan, M. T. Quasim, N. S. Alghamdi, and M. Y. Khan, "A secure framework for authentication and encryption using improved ECC for IoT-based medical sensor data," *IEEE Access*, vol. 8, pp. 52018–52027, 2020.
[23] R. Imam, Q. M. Areeb, A. Alturki, and F. Anwer, "Systematic and critical review of RSA based public key cryptographic schemes: Past and present status," *IEEE Access*, vol. 9, pp. 155949–155976, 2021.
[24] N. Sen, R. Dantu, J. Vempati, and M. Thompson, "Performance analysis of elliptic curves for real-time video encryption," in *Proc. Nat. Cyber Summit (NCS)*, Jun. 2018, pp. 64–71.
[25] S. C. Iyer, R. R. Sedamkar, and S. Gupta, "A novel idea on multimedia encryption using hybrid crypto approach," *Proc. Comput. Sci.*, vol. 79, pp. 293–298, Jan. 2016. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050916001691
[26] P. R. Vijayalakshmi and K. B. Raja, "Performance analysis of RSA and ECC in identity-based authenticated new multiparty key agreement protocol," in *Proc. Int. Conf. Comput., Commun. Appl.*, Feb. 2012, pp. 1–5.
[27] Z. Chen and G. Ye, "An asymmetric image encryption scheme based on hash SHA-3, RSA and compressive sensing," *Optik*, vol. 267, Oct. 2022, Art. no. 169676. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0030402622009627
[28] R. Hegde and S. Jagadeesha, "An optimal modified matrix encoding technique for secret writing in MPEG video using ECC," *Comput. Standards Interfaces*, vol. 48, pp. 173–182, Nov. 2016.
[29] S. H. Murad and K. H. Rahouma, "Implementation and performance analysis of hybrid cryptographic schemes applied in cloud computing environment," *Proc. Comput. Sci.*, vol. 194, pp. 165–172, Jan. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050921021116
[30] S. K. Ghosh, S. Rana, A. Pansari, J. Hazra, and S. Biswas, "Hybrid cryptography algorithm for secure and low cost communication," in *Proc. Int. Conf. Comput. Sci., Eng. Appl. (ICCSEA)*, Mar. 2020, pp. 1–5.
[31] J. Zhang and X. Gao, "A hybrid encryption scheme for scalable video coding based on H.264," in *Proc. 5th Int. Conf. Intell. Netw. Collaborative Syst.*, Sep. 2013, pp. 708–711.
[32] L. Yu, Z. Wang, and W. Wang, "The application of hybrid encryption algorithm in software security," in *Proc. 4th Int. Conf. Comput. Intell. Commun. Netw.*, Nov. 2012, pp. 762–765.
[33] M. A. Khan, K. K. Mishra, N. Santhi, and J. Jayakumari, "A new hybrid technique for data encryption," in *Proc. Global Conf. Commun. Technol. (GCCT)*, Apr. 2015, pp. 925–929.
[34] C. L. Chowdhary, P. V. Patel, K. J. Kathrotia, M. Attique, K. Perumal, and M. F. Ijaz, "Analytical study of hybrid techniques for image encryption and decryption," *Sensors*, vol. 20, no. 18, p. 5162, Sep. 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/18/5162
[35] J. Dave and M. Gayathri, "Hybrid encryption algorithm for storing unimodal biometric templates in cloud," in *Inventive Communication and Computational Technologies*, G. Ranganathan, X. Fernando, and F. Shi, Eds. Singapore: Springer, 2022, pp. 251–266.
[36] M. Hamdi, J. Miri, and B. Moalla, "Hybrid encryption algorithm (HEA) based on chaotic system," *Soft Comput.*, vol. 25, no. 3, pp. 1847–1858, Feb. 2021, doi: 10.1007/s00500-020-05258-z.
[37] P. Yu, N. Zhang, S. Zhang, and Q. Wang, "Security mechanism of video content integrated broadcast control platform under triple play," in *Proc. 10th Int. Congr. Image Signal Process., Biomed. Eng. Informat. (CISP-BMEI)*, Oct. 2017, pp. 1–5.
[38] D. Zhu, J. Zheng, H. Zhou, J. Wu, N. Li, and L. Song, "A hybrid encryption scheme for quantum secure video conferencing combined with blockchain," *Mathematics*, vol. 10, no. 17, p. 3037, Aug. 2022. [Online]. Available: https://www.mdpi.com/2227-7390/10/17/3037
[39] K. M. Hosny, M. A. Zaki, N. A. Lashin, and H. M. Hamza, "Fast colored video encryption using block scrambling and multi-key generation," *Vis. Comput.*, pp. 1–32, Nov. 2022. [Online]. Available: https://link.springer.com/article/10.1007/s00371-022-02711-y
[40] A. Alarifi, S. Sankar, T. Altameem, K. Jithin, M. Amoon, and W. El-Shafai, "A novel hybrid cryptosystem for secure streaming of high efficiency H.265 compressed videos in IoT multimedia applications," *IEEE Access*, vol. 8, pp. 128548–128573, 2020.



YOUCEF FOUZAR received the B.Sc. and M.Sc. degrees in electrical engineering from the Polytechnic University of Kharkov, Ukraine, in 1995 and 1996, respectively, and the Ph.D. degree in electrical engineering from the École Polytechnique de Montréal, Canada, in 2004. His research interests include hardware and software development involving cryptography and data privacy.

AHMED LAKHSSASSI (Senior Member, IEEE) received the B.Eng. and M.Sc. degrees in electrical engineering from the Université du Québec à Trois-Rivières (UQTR), Trois-Rivières, QC, Canada, in 1988 and 1990, respectively, and the Ph.D. degree in energy and material sciences from the Institut national de la recherche scientifique (INRS)-Énergie, Montreal, in 1995. In 1995, he became a Professor of electro-thermo-mechanical aspects with the Natural Sciences and Engineering Research Council of Canada (NSERC)-Hydro-Quebec Industrial Research Chair, Department of Electrical Engineering, UQTR. Since 1998, he has been with the Université du Québec en Outaouais (UQO), Gatineau. He is currently a Titular Professor and responsible for the Laboratory of Advanced Microsystem Engineering (LIMA), developing IP core and embedded algorithms for microsystems thermo-mechanical sensors dedicated to thermal peak detection. He is the author/coauthor of more than 240 scientific publications and research reports and a thesis advisor of 90 graduate and undergraduate students who completed their studies. His research interests include bio-heat thermal modeling and heat transfer mechanisms in biological tissues for thermal therapeutic practices. He is a Regular Member of the Strategic Alliance in Microsystems of Québec (ReSMiQ), the largest research center in microelectronics funded by the Government of Québec.

M. RAMAKRISHNA (Member, IEEE) received the B.E. degree in computer science and engineering from Visvesvaraya Technological University, India, in 2008, and the M.Tech. degree in network engineering and the Ph.D. degree in multimedia communication from the Manipal Academy of Higher Education, Manipal, India, in 2010 and 2019, respectively.

He worked as an Early-Stage Researcher at the Intel Visual Computing Institute, Saarland University, Germany. He was a Research Associate at the Department of Computer Science, University of Bath, U.K. He worked as a Guest Researcher at Telecommunication Laboratories, Saarland University. His research interests include image processing, virtual reality, video coding, video adaptation, network protocols, routing algorithms, and software-defined networks.