Département d'informatique et d'ingénierie
Laboratoire d'ingénierie des microsystèmes avancés (LIMA)

Thèse présentée en vue de l'obtention du grade de

DOCTORAT

en Science et Technologie de l'Information

Étude et conception d'une plateforme pour un partage sécurisé du contenu vidéo

---

Présenté par : Youcef Fouzar

Directeur de thèse : Professeur Ahmed Lakhssassi, Ph.D.

Date: Mai 2022

Résumé

Les vidéos sont aujourd'hui une source majeure de moyens de communication, tant à des fins professionnelles que personnelles. Leur nombre augmente de manière exponentielle et la confidentialité de leur contenu est devenue un enjeu majeur pour leur acquisition, leur transmission, leur stockage et leur affichage.

Le moyen le plus efficace de sécuriser le contenu vidéo est d'utiliser des techniques de cryptage. Cela permet d'éviter le piratage du contenu vidéo au détriment de la complexité accrue de la lecture des vidéos cryptées.

Pour lire une vidéo cryptée, il faut d'abord la décrypter. Cela signifie que le stockage de la vidéo non cryptée est inévitable. Ce stockage peut être permanent, en décryptant l'intégralité de la vidéo une fois pour plusieurs lectures, ou temporaire, à chaque lecture de la vidéo. Cette situation peut donner lieu à une faille de sécurité majeure, avec le risque de fuite de contenu non crypté à partir de la vidéo en claire, qui pourrait être copiée et redistribuée sans autorisation. Pour résoudre ce problème, il est essentiel de créer une plateforme de gestion de vidéos cryptées afin de garantir leur confidentialité.

Pour plus de sécurité, la vidéo est divisée en plusieurs petits morceaux, et une clé de décryptage unique est créée pour chaque morceau. Nous avons également adopté une cryptographie hybride utilisant les algorithmes AES et ECC en plus de l'algorithme RSA utilisé dans la création des identifiants vidéo.

La génération des clés elle-même est gérée au sein de la plateforme proposée, sans qu'il soit nécessaire de transmettre les clés comme dans les usages conventionnels. En outre, la génération de clés est basée sur une nouvelle méthode qui prend en compte les attributs du récepteur pour créer des clés de décryptage uniques au sein de la plateforme. Cette méthode

renforce la sécurité et réduit le risque d'interception des clés pendant la transmission entre l'expéditeur et le destinataire dans les systèmes classiques.

Mots-clés : plateforme vidéo, cryptage, décryptage, ECC, AES et RSA

Abstract

Videos are today a major source of means of communication, for both professional and personal purposes. Their number is growing exponentially, and the confidentiality of their content has become a major issue for their acquisition, transmission, storage and display.

The most effective way to secure video content is to use encryption techniques. This prevents piracy of video content at the expense of the added complexity of playing encrypted videos.

To play an encrypted video, it must first be decrypted. This means that storage of the unencrypted video is unavoidable. This storage can be permanent, by decrypting the entire video once for multiple playbacks, or temporary, each time the video is played back. This could lead to a major security breach, with the risk of unencrypted content leaking from the clear video, which could be copied and redistributed without authorization. To solve this problem, it is essential to create a platform for managing encrypted videos, in order to guarantee their confidentiality.

For added security, the video is divided into several smaller chunks, and a unique encryption key is created for each chunk. Furthermore, a hybrid cryptography is adopted using AES and ECC algorithms. In addition, an RSA algorithm is used to create video identifiers.

Key generation is managed within the proposed platform, without the need to transmit the keys as in conventional solutions. In addition, key generation is based on a novel method that considers receiver attributes to create unique decryption keys within the platform. This method enhances security and reduces the risk of keys being intercepted during transmission between sender and receiver in conventional systems.

Keywords: video platform, encryption, decryption, ECC, AES and RSA

Table de matières

Liste des Figures

Dédicace

A la mémoire de mon père Amar Fouzar

A cette source de tendresse, de patience et de générosité, à ma mère !

A mes chers enfants : Amine, Younes, Adam, Alaa et Omar

Je souhaite sincèrement que le temps et les efforts consacrés à cette thèse leur servent d'inspiration pour un avenir meilleur

A mes frères (Rabah, Kamel et Mourad) et à mes sœurs (Leila, Zoubida, Nadia, Mounia, Samia et Amina) pour leur soutient et support

Remerciements

Je souhaite premièrement remercier mon directeur de thèse, Monsieur Ahmed Lakhssassi, Professeur au département d'informatique et d'ingénierie à Université du Québec en Outaouais (UQO). Professeur Lakhssassi et aussi le créateur et dirigent du Laboratoire d'ingénierie des microsystèmes avancés (LIMA). Professeur Lakhssassi m'a encadré tout au long de cette thèse. Qu'il soit aussi remercié pour sa gentillesse, sa disponibilité permanente et pour les nombreux encouragements qu'il m'a prodiguée.

Je remercie aussi Monsieur Emmanuel Kengne, Professeur au département d'informatique et d'ingénierie à Université du Québec en Outaouais (UQO) et Co-directeur cette thèse.

Je tiens à remercier Madame Ilham Benyahia, Professeure au département d'informatique et d'ingénierie à Université du Québec en Outaouais (UQO) pour l'honneur qu'elle m'a fait en acceptant d'être la présidente de mon jury de thèse. Je tiens à l'assurer de ma profonde reconnaissance pour l'intérêt qu'elle porte à ce travail.

J'exprime ma gratitude à Monsieur Adam W. Skorek, Professeur au département de génie électrique et génie informatique à l'Université de Université du Québec à Trois-Rivières (UQTR) pour l'honneur qu'il m'a fait en acceptant d'être membre évaluateur externe de cette thèse, pour le temps consacré à la lecture de cette thèse, et pour les suggestions et les remarques judicieuses qu'il m'a indiquées.

J'adresse tous mes remerciements à Madame Nadia Baaziz, Professeure au département d'informatique et d'ingénierie à Université du Québec en Outaouais (UQO) pour avoir acceptée d'être le membre évaluateur interne de cette thèse. Je suis reconnaissant pour toutes les questions et les commentaires qu'elle a formulés afin d'améliorer cette thèse.

Introduction générale

Aperçu général

La diffusion en continu de médias en temps réel est devenue une commodité en raison des progrès et des développements rapides de notre infrastructure Internet et des applications qui pilotent ces technologies [1], [2]. Le streaming media permet de diffuser en continu des données média, telles que de l'audio ou de la vidéo, sur Internet. Le contenu est présenté à l'utilisateur final avant qu'il n'ait été entièrement téléchargé. En raison de la popularité croissante de la vidéoconférence, des services de télévision sur le Web, de l'apprentissage en ligne, de la télémédecine et des entreprises populaires sur Internet comme YouTube et Netflix, des services de diffusion en continu de médias en direct sont proposés aux entreprises et aux particuliers [3], [4]. Par conséquent, le partage du trafic Internet est plus important. De plus, l'Internet est un réseau décentralisé ; n'importe qui peut se connecter de n'importe où et partager n'importe quelle donnée multimédia [5], [6].

L'augmentation du trafic vidéo des services OTT (Over-The-Top) [5]-[7] et d'autres applications similaires a suscité des préoccupations importantes en matière de sécurité et de confidentialité. Les consommateurs et les producteurs ont été confrontés au partage illégal de contenu multimédia et à des vidéos piratées de données sensibles. Ces perturbations concernent principalement les données liées à la télémédecine et au streaming vidéo en temps réel. Par conséquent, une technique de cryptographie appropriée est nécessaire pour gérer les problèmes de communication vidéo. Les défis les plus critiques sur lesquels il faut se concentrer sont l'authentification, le cryptage et la gestion des clés [8], [9].

Les mesures de sécurité suivantes sont utilisées pour la protection du contenu : (i) filigrane judiciaire pour empêcher l'acquisition du contenu pendant le rendu ; (ii) environnement de calcul de confiance pour empêcher l'accès pendant le décodage ; et (iii) cryptage pour interdire l'accès au contenu pendant le transit. L'introduction de la vidéo de nouvelle génération et la popularité croissante des dispositifs embarqués pour la consommation de contenu exigent de nouvelles stratégies de protection du contenu qui reposent moins sur le matériel.

En général, les applications de streaming vidéo utilisent les méthodes suivantes pour sécuriser le contenu vidéo sur une plateforme de streaming. Méthodes de diffusion vidéo en continu sécurisées : Techniques de cryptographie, diffusion HTTPS, Paywall de paiement crypté SSL/TLS, vidéo protégée par mot de passe, restrictions géographiques (IP), restrictions de référence et centre de données vidéo et CDN sécurisés.

De nombreuses techniques de cryptographie sont mises en œuvre dans les applications pour améliorer la sécurité des données multimédia. La cryptographie à clé symétrique et asymétrique est la technique disponible pour assurer la sécurité des communications. La cryptographie à clé symétrique est la plus utilisée dans les communications multimédia [10]-[12]. Norme de cryptage avancée (Advanced Encryption Standard) (AES) [13] est le cryptage symétrique le plus efficace et le plus utilisé. Les sites Web et les navigateurs Internet utilisent l'AES 128 bits pour assurer la sécurité des communications sur Internet. La méthode de gestion des clés a été problématique dans cette procédure en raison de la mise en œuvre du protocole de transport en temps réel sécurisé (Secure Real-time Transport Protocol) (SRTP). Comme le réseau est décentralisé, la gestion des clés devient un défi. Dans de nombreuses techniques, la clé et les algorithmes ne peuvent pas être divisés efficacement pour améliorer la sécurité de l'Internet. Par conséquent, de nombreux travaux de recherche sont en cours pour réaliser des techniques de cryptographie à clé asymétrique [14],

[15]. Les méthodes de cryptographie asymétrique telles que Rivest-Shamir-Adleman (RSA) [16] et Elliptic Curve Cryptography (ECC) [17]-[19] ont été explorées et gagnent en popularité pour surmonter les défis de la cryptographie symétrique. Dans une application de diffusion vidéo en continu, les données vidéo sont divisées en plusieurs morceaux, puis diffusées à l'aide de protocoles de diffusion en continu. La plupart des méthodes traditionnelles de cryptage et de décryptage utilisent la cryptographie à clé symétrique, mais les méthodes d'échange de clés dans ces techniques entraînent des failles de sécurité. Par conséquent, les techniques de cryptographie à clé asymétrique permettent de renforcer la sécurité du contenu vidéo diffusé en continu sur Internet.

Les attaques dites "man-in-the-middle" sont protégées par HTTPS [20]-[22]. Ces attaques sont plutôt typiques du streaming vidéo, notamment lorsque les utilisateurs accèdent au contenu vidéo sur des réseaux ouverts dans les bibliothèques, les cafés et les écoles. La transmission des données au spectateur peut être interceptée par des pirates utilisant des failles dans ces réseaux ouverts. La diffusion HTTPS utilise le cryptage HLS [23], [24] pour dissimuler la connexion de l'utilisateur avec le site Web et empêcher ce type d'attaque à l'aide de certificats numériques et de clés de cryptage. Toute connexion entre le serveur qui transmet les films et le spectateur est protégée par une couche de cryptage sécurisée lorsqu'on utilise le protocole HTTPS. Elle offre au matériel vidéo une protection supplémentaire. La transmission par HTTPS est une excellente solution pour protéger les vidéos sur Internet.

La diffusion en direct protégée par mot de passe [23], [24] est une approche fantastique pour tenir les utilisateurs non autorisés à l'écart du matériel vidéo, même si elle semble simple. Voici comment fonctionne la vidéo protégée par mot de passe : Le programme crée d'abord un mot de passe pour les vidéos sélectionnées. Ensuite, les spectateurs doivent saisir le bon mot de

passe pour pouvoir regarder. S'ils ne connaissent pas le mot de passe, ils ne peuvent pas voir la vidéo. La protection par mot de passe est une mesure de sécurité rapide et efficace, idéale pour les films d'anticipation, l'usage interne de l'entreprise, les évaluations des clients, etc. Il est également essentiel de se rappeler que ce n'est pas la seule ligne de défense. Les mots de passe pourraient être exposés sur des forums Internet, en fonction de la popularité de votre entreprise. C'est pourquoi de nombreux diffuseurs modifient périodiquement leurs mots de passe pour masquer les accès non autorisés. Associé à d'autres mesures de sécurité, le streaming vidéo protégé par mot de passe est un outil puissant pour assurer la sécurité de votre matériel.

Les limitations géographiques ou IP [25]-[27] peuvent être une option si les téléspectateurs se trouvent dans des régions ou des pays particuliers. Grâce à cette technologie, vous pouvez prévenir le piratage en empêchant certaines zones géographiques de voir vos films. Certaines zones sont celles où le piratage est le plus répandu. Les géo-restrictions permettent d'établir facilement une "liste noire" de certains pays. Toutefois, cela empêchera les spectateurs de ces pays de regarder vos films. De nombreuses sociétés visent à cibler les spectateurs d'un ou deux pays avec leurs films. Les limitations géographiques permettent donc de mettre sur une liste blanche certains pays et sur une liste noire tous les autres. La liste blanche fonctionne grâce aux adresses IP, qui sont associées à des lieux. Elle n'est pas totalement infaillible, comme d'autres mesures de sécurité, mais elle peut offrir une sécurité importante. Il peut s'agir d'une technique de sécurité pour le streaming si le public cible est situé dans un pays spécifique.

Les restrictions de référence fonctionnent en établissant une liste blanche des domaines spécifiques qui sont autorisés à lire notre contenu vidéo [28]. Ces domaines comprennent souvent le site web et les sites affiliés. Lorsqu'elle est activée, cette fonction utilise un réseau de jetons de sécurité numériques pour vérifier périodiquement le serveur. La lecture de la vidéo est

immédiatement bloquée s'il s'avère qu'elle est intégrée à un site Web non approuvé. La vidéo ne peut pas être diffusée par quiconque tente de l'intégrer sur un autre site web.

Un CDN est un réseau informatique qui utilise un logiciel d'équilibrage de charge sophistiqué pour envoyer des vidéos et d'autres données aussi rapidement que possible dans le monde entier [29], [30]. Le matériel vidéo se charge rapidement avec peu de latence ou de mise en mémoire tampon, grâce à un CDN de diffusion en direct. L'utilisation d'une plateforme vidéo en ligne intégrée à un CDN permet de se défendre contre toute une série de menaces. Cela inclut les attaques "DDoS", qui tentent de faire tomber un site web en l'inondant de trafic. Ce type d'attaque de cybersécurité est principalement rendu inutile par l'utilisation d'un CDN. En outre, les CDN comprennent une protection contre les pannes matérielles. Ils offrent une redondance intégrée, garantissant la sécurité et le chargement rapide de votre matériel.

La capacité de ces technologies à assurer la sécurité dans des environnements dynamiques et autonomes est limitée. Étant donné que les méthodes ci-dessus utilisent l'échange de clés pour garantir la sécurité, il est nécessaire de protéger la clé pendant la transmission. Il faut une plateforme qui offre une cryptographie à clés multiples avec des frais généraux de gestion des clés minimaux.

Problématique et objectif

De nos jours, les vidéos sont des sources majeures de communication à des fins professionnelles ou personnelles. Leur nombre croît de manière exponentielle et la confidentialité de leur contenu est devenue un problème majeur pour leur acquisition, leur transmission, leur stockage et leur affichage.

La lecture de la vidéo cryptée requiert que la vidéo cryptée soit décryptée d'abord. Ceci implique un stockage de la vidéo en format claire est inévitable. Ce stockage pourrait être permanent en décryptons la totalité de la vidéo une fois pour plusieurs lectures ou temporaire à chaque lecture de la vidéo. Cette situation pourra engendrée un problème majeur de sécurité avec le risque de fuite du contenu de la vidéo en claire qui pourra être volée et redistribuée sans autorisation. Afin de résoudre ce problème, une plateforme de cryptage et de gestion vidéo qui permet d'assurer la confidentialité est vital.

L'objectif principal de cette thèse est de développer une plateforme de sécurisation du contenu vidéos. Cette plateforme permet au contenu vidéo d'être crypté, partagé et visualisé par le receveur et stockée toujours dans la forme cryptée sans permettre de laisser la vidéo en format clair dans la mémoire de l'appareil.

La plateforme chargée de sauvegarder les vidéos cryptées en local doit être en mesure de les lire uniquement au sein de la plateforme. L'utilisateur peut crypter les vidéos et les stocker dans un dossier local. Cependant, la stratégie de la lecture des vidéos cryptées pourra s'inspirer des solutions existantes pour sécuriser la diffusion en direct en ligne en streaming. A ce jour, il n'existe pas de moyen permettant de lire, en sécurité, les vidéos cryptées hors ligne sans laisser des parties de la vidéo décryptées dans la mémoire.

Organisation de la thèse

Cette thèse est une présentation d'un travail original dans le domaine du cryptage vidéo en utilisant la technique dynamique à clés multiples proposée pour tenter de construire un système de cryptage sûr et robuste. La technique pourrait être appliquée à tout type et format de vidéo. La thèse se concentre davantage sur les aspects de conception de circuit et la réalisation matérielle du système de cryptage en plus de présenter les résultats de l'analyse de sécurité. De plus, les résultats sont comparés avec des systèmes similaires rapportés dans la littérature et la supériorité des performances globales est démontrée. Une implémentation sur une plateforme Android est réalisée aussi et démontre que cette technique proposée peut être utilisée dans les appareils mobiles intelligents.

Le reste de cette thèse est organisé comme suit : Le chapitre un présente la méthodologie de cette thèse avec description de la problématique. Le chapitre deux présente le cryptage vidéo avec une revue de littérature. Le chapitre trois présente le cadre de cryptage vidéo en parallèle avec sa mise en œuvre numérique, puis analyse la sécurité du système de cryptage proposé. Le chapitre quatre montre les résultats expérimentaux de la réalisation de la plateforme. Enfin, la conclusion résume le travail présenté dans cette thèse et fournit quelques suggestions pour le travail futur et les recherches ouvertes.

Référence

[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 Transactions on Broadcasting*, vol. 64, no. 1, pp. 11–25, 2018, doi: 10.1109/TBC.2017.2781146.

[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, doi: 10.1155/2021/8875700.

[3] D. Shamsimukhametov, M. Liubogoshchev, E. Khorov, and I. F. Akyildiz, “YouTube, Netflix, Web dataset for Encrypted Traffic Classification.” IEEE Dataport, 2021. doi: 10.21227/s7x7-wd58.

[4] 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,” Sep. 2022, doi: 10.6084/m9.figshare.19096823.v2.

[5] “Cisco Annual Internet Report - Cisco Annual Internet Report (2018–2023) White Paper,” *Cisco*. [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,” *Cisco*. [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,” *Proceedings of the 7th Conference on Emerging Networking EXperiments and Technologies, CoNEXT’11*, 2011, doi: 10.1145/2079296.2079321.

[8] X. Huang, D. Arnold, T. Fang, and J. Saniie, “A Chaotic-based Encryption/Decryption System for Secure Video Transmission,” in *2021 IEEE International Conference on Electro Information Technology (EIT)*, 2021, pp. 369–373. doi: 10.1109/EIT51626.2021.9491868.

[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, no. 1, Dec. 2008.

[10] A. Murtaza, S. J. Hussain Pirzada, and L. Jianwei, “A New Symmetric Key Encryption Algorithm with Higher Performance,” in *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2019, pp. 1–7. doi: 10.1109/ICOMET.2019.8673469.

[11] S. Kansal and M. Mittal, “Performance evaluation of various symmetric encryption algorithms,” in *2014 International Conference on Parallel, Distributed and Grid Computing*, 2014, pp. 105–109. doi: 10.1109/PDGC.2014.7030724.

[12] S. Kumar, M. S. Gaur, P. Sagar Sharma, and D. Munjal, “A Novel Approach of Symmetric Key Cryptography,” in *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, 2021, pp. 593–598. doi: 10.1109/ICIEM51511.2021.9445343.

[13] M. Dworkin *et al.*, “Advanced Encryption Standard (AES).” Federal Inf. Process. Stds. (NIST FIPS), National Institute of Standards and Technology, Gaithersburg, MD, Sep. 2001. doi: https://doi.org/10.6028/NIST.FIPS.197.

[14] Y. Shen, Z. Sun, and T. Zhou, “Survey on Asymmetric Cryptography Algorithms,” in *2021 International Conference on Electronic Information Engineering and Computer Science (EIECS)*, 2021, pp. 464–469. doi: 10.1109/EIECS53707.2021.9588106.

[15] S. Kumar, B. K. Singh, Akshita, S. Pundir, S. Batra, and R. Joshi, “A survey on Symmetric and Asymmetric Key based Image Encryption,” in *2nd International Conference on Data,*

Engineering and Applications (IDEA), 2020, pp. 1–5. doi: 10.1109/IDEA49133.2020.9170703.

[16] R. L. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," *Commun ACM*, vol. 21, pp. 120–126, 1978.

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

[18] A. J. Menezes and S. A. Vanstone, "Elliptic curve cryptosystems and their implementation," *Journal of Cryptology 1993 6:4*, vol. 6, no. 4, pp. 209–224, Sep. 1993, doi: 10.1007/BF00203817.

[19] N. Koblitz, A. Menezes, and S. Vanstone, "The State of Elliptic Curve Cryptography," *Designs, Codes and Cryptography 2000 19:2*, vol. 19, no. 2, pp. 173–193, 2000, doi: 10.1023/A:1008354106356.

[20] H. Xu, Z. Chen, R. Chen, and J. Cao, "Live streaming with content centric networking," *Proceedings of the International Conference on Networking and Distributed Computing, ICNDC*, pp. 1–5, 2012, doi: 10.1109/ICNDC.2012.9.

[21] N. Prabhu, D. Naik, and F. Anwar, "Trusted Video Streaming on Edge Devices," 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events, PerCom Workshops 2021, pp. 655–660, Mar. 2021, doi: 10.1109/PERCOMWORKSHOPS51409.2021.9431058.

[22] A. M. Elshamy, M. A. Abdelghany, A. Q. Alhamad, H. F. A. Hamed, H. M. Kelash, and A. I. Hussein, "Secure Implementation for Video Streams Based on Fully and Permutation Encryption Techniques," *2017 International Conference on Computer and Applications, ICCA 2017*, pp. 50–55, Oct. 2017, doi: 10.1109/COMAPP.2017.8079738.

[23] A. K. Chaurasiya and P. Yadav, "Live Broadcast Data Processing Security by Software Based Encryption," *2019 15th International Conference on Information Processing:*

Internet of Things, ICINPRO 2019 - Proceedings, Dec. 2019, doi: 10.1109/ICINPRO47689.2019.9092204.

[24] J. Zhou and C. M. Pun, "Personal Privacy Protection via Irrelevant Faces Tracking and Pixelation in Video Live Streaming," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1088–1103, 2021, doi: 10.1109/TIFS.2020.3029913.

[25] R. E. Harang and W. J. Glodek, "Identification of anomalous network security token usage via clustering and density estimation," *2012 46th Annual Conference on Information Sciences and Systems, CISS 2012*, 2012, doi: 10.1109/CISS.2012.6310829.

[26] S. T. Li and X. Wang, "Ad hoc network security with geographical aids," *Conference Proceeding - IEEE International Conference on Networking, Sensing and Control*, vol. 1, pp. 474–479, 2004, doi: 10.1109/ICNSC.2004.1297484.

[27] H. Maziku, S. Shetty, K. Han, and T. Rogers, "Enhancing the classification accuracy of IP geolocation," *Proceedings - IEEE Military Communications Conference MILCOM*, 2012, doi: 10.1109/MILCOM.2012.6415842.

[28] L. Zhang, Z. Wei, W. Ren, X. Zheng, K. K. R. Choo, and N. N. Xiong, "SIP: An Efficient and Secure Information Propagation Scheme in E-Health Networks," *IEEE Trans Netw Sci Eng*, vol. 8, no. 2, pp. 1502–1516, Apr. 2021, doi: 10.1109/TNSE.2021.3063174.

[29] M. Ghaznavi, E. Jalalpour, M. A. Salahuddin, R. Boutaba, D. Migault, and S. Preda, "Content Delivery Network Security: A Survey," *IEEE Communications Surveys and Tutorials*, vol. 23, no. 4, pp. 2166–2190, 2021, doi: 10.1109/COMST.2021.3093492.

[30] B. Zhou and X. Pan, "Cookie-based CDN security authorization design," *2011 International Conference on Internet Technology and Applications, iTAP 2011 - Proceedings*, 2011, doi: 10.1109/ITAP.2011.6006202.

Chapitre 1 : Méthodologie

Nous ciblons une contrainte majeure qui est la sécurisation et la confidentialité des données partagées en ligne. Notre proposition s'applique au contenu vidéo partagé en ligne. La majorité des entreprises ont des données à protéger, qui peuvent inclure des vidéos. Il existe de nombreux logiciels et applications pouvant chiffrer ces vidéos. Le problème majeur est que la clé utilisée pour le chiffrement n'est pas protégée une fois qu'elle a été créée.

Si une vidéo doit être partagée avec une certaine personne, nous nous assurons que seule cette personne est capable de déchiffrer ou de lire cette vidéo. Nous proposons aussi d'utiliser la cryptographie asymétrique pour les nombreux avantages qu'elle offre. Si la vidéo doit être partagée avec un groupe d'individus, alors la clé ne sera pas distribuée mais centralisée, sans partage, dans le serveur et seuls les utilisateurs privilégiés pourront décrypter ces vidéos. Dans le but d'avoir un contrôle total sur les applications et les utilisateurs, nous allons stocker toutes les clés liées à cette application sur un serveur en ligne. Nous planifions de concevoir un prototype de telle sorte qu'il utilise notre algorithme de cryptage personnalisé et prend 10 fois moins de temps par rapport au temps pris par une application Androide générale avec un algorithme personnalisé.

Pour notre méthode nous allons utiliser un chiffrement AES, Lorsque nous créons un compte pour utilisateur, nous générons une clé publique et une clé privée et nous les stockons en ligne. Lorsqu'un utilisateur essaie de chiffrer une vidéo spécialement pour un autre utilisateur, nous lui demandons de saisir le nom de réception. A ce moment, nous récupérons la clé publique du récepteur et l'utilisons pour chiffrer la clé générée pour AES. Maintenant que la clé publique du récepteur est utilisée dans le processus de chiffrement de la clé AES, personne d'autre que le récepteur ne peut déchiffrer la clé, même si le destinataire partage la vidéo et la clé avec d'autres. Lorsque le récepteur essaie de lire la vidéo avec une clé donnée, nous déchiffrons d'abord la clé en

utilisant la clé privée du récepteur, puis nous lisons la vidéo avec la clé déchiffrée. Pour réaliser cet objectif nous allons implémenter une méthode qui permettra de partager sur une plateforme point à point (Peer to Peer) ou point à groupe (Peer to Group) un contenu vidéo crypté sans partager le fichier original de la vidéo. Cette méthode pourrait faire l'objet d'un article scientifique ou même d'un brevet en y ajoutant une méthode unique de cryptage des fichiers vidéo proposée dans cette thèse. Cette méthode qui a été implémentée sur une plateforme Android a permis de démontrer le concept innovant de cette solution. De plus, la solution complète sera présentée dans le manuscrit de la thèse.

Chapitre 2 : État de l’art sur le cryptage des données

Problématique et objectif

De nos jours, les vidéos sont des sources majeures de communication à des fins professionnelles ou personnelles. Leur nombre augmente de façon exponentielle et la confidentialité de leur contenu est devenue un enjeu majeur pour leur acquisition, leur transmission, leur stockage et leur affichage.

La lecture d'une vidéo cryptée exige que cette dernière soit d'abord décryptée. Cela implique que le stockage de la vidéo en clair est inévitable. Ce stockage peut être permanent en décryptant l'ensemble de la vidéo une fois pour plusieurs lectures ou temporaire à chaque fois que la vidéo est lue. Cette situation pourrait entraîner un problème de sécurité majeur avec le risque de fuite du contenu vidéo en clair qui pourrait être volé et redistribué sans autorisation. Pour résoudre ce problème, une plateforme de cryptage et de gestion de la vidéo qui garantit la confidentialité est indispensable.

L'objectif principal de cette thèse est de développer une plateforme pour sécuriser le contenu vidéo. Cette plateforme permet au contenu vidéo d'être crypté, partagé et visualisé par le destinataire et stocké toujours sous la forme cryptée sans permettre à la vidéo d'être laissée en clair dans la mémoire de l'appareil.

La plateforme responsable de la sauvegarde locale des vidéos cryptées doit pouvoir les lire uniquement à l'intérieur de la plateforme. L'utilisateur peut crypter les vidéos et les stocker dans un dossier local. Cependant, la stratégie de lecture des vidéos cryptées peut s'appuyer sur les solutions existantes pour sécuriser le streaming en ligne en direct. À ce jour, il n'existe aucun

moyen de lire en toute sécurité des vidéos cryptées hors ligne sans laisser en mémoire des parties de la vidéo décryptée.

Introduction

La diffusion en continu de médias en temps réel est devenue une commodité en raison des progrès et des développements rapides de notre infrastructure Internet et des applications qui pilotent ces technologies [1] [2]. Le média en continu est la livraison continue de données média, telles que l'audio ou la vidéo, sur Internet, où le contenu est présenté à l'utilisateur final avant d'avoir été complètement téléchargé. En raison de la popularité croissante de la vidéoconférence, des services de télévision sur le Web, de l'apprentissage en ligne, de la télémédecine et des entreprises populaires basées sur l'Internet comme YouTube et Netflix, des services de diffusion en continu de médias en direct sont proposés aux entreprises et aux particuliers [3] [4]. Par conséquent, le partage du trafic Internet est plus important. En outre, l'internet est un réseau décentralisé ; par conséquent, n'importe qui peut se connecter de n'importe où et partager n'importe quelle donnée médiatique [5].

L'augmentation du trafic vidéo des services OTT (Over-The-Top) [6] [7] et d'autres applications similaires a suscité de vives inquiétudes en matière de sécurité et de confidentialité. Les consommateurs et les producteurs ont été confrontés au partage illégal de contenu multimédia et à des vidéos piratées de données sensibles. Ces perturbations concernent principalement les données liées à la télémédecine et au streaming vidéo en temps réel. Par conséquent, une technique de cryptographie appropriée est nécessaire pour gérer les problèmes de communication vidéo. Les défis les plus critiques sur lesquels il faut se concentrer sont l'authentification, le cryptage et la gestion des clés [8] [9].

De nombreuses techniques de cryptographie sont mises en œuvre dans les applications pour améliorer la sécurité des données multimédia. La cryptographie à clé symétrique et asymétrique est la technique disponible pour assurer la sécurité des communications. La cryptographie à clé symétrique est la plus utilisée dans les communications multimédia [10] [11] [12]. Advanced Encryption Standard (AES) [13] est le cryptage symétrique le plus efficace et le plus utilisé. Les sites Web et les navigateurs Internet utilisent l'AES 128 bits pour assurer la sécurité des communications sur Internet. La méthode de gestion des clés a été problématique dans cette procédure en raison de la mise en œuvre du protocole de transport en temps réel sécurisé (SRTP). Comme le réseau est décentralisé, la gestion des clés devient un défi. Dans de nombreuses techniques, la clé et les algorithmes ne peuvent pas être divisés efficacement pour améliorer la sécurité de l'Internet. Par conséquent, de nombreux travaux de recherche sont en cours pour réaliser des techniques de cryptographie à clé asymétrique [14] [15]. Les méthodes de cryptographie asymétrique telles que Rivest-Shamir-Adleman (RSA) [16] et la cryptographie à courbe elliptique (ECC)

[17] [18] [19] ont été explorées et gagnent en popularité pour surmonter les défis de la cryptographie symétrique. Dans une application de diffusion vidéo en continu, les données vidéo sont divisées en plusieurs morceaux, puis diffusées à l'aide de protocoles de diffusion en continu. La plupart des méthodes traditionnelles de cryptage et de décryptage utilisent la cryptographie à clé symétrique, mais les méthodes d'échange de clés dans ces techniques entraînent des problèmes de sécurité. Par conséquent, les techniques de cryptographie à clé asymétrique permettent de renforcer la sécurité du contenu vidéo diffusé en continu sur Internet. Cet article a développé une nouvelle méthode qui utilise la cryptographie à clé asymétrique pour crypter des morceaux de vidéo. La contribution de ce travail consiste à concevoir et à développer une nouvelle technique

de cryptage à clés multiples basée sur des équations et une méthode de génération de clé de décryptage basée sur des attributs vidéo.

Ce travail vise à développer une technique de cryptographie à clés multiples pour les applications de streaming vidéo. Les caractéristiques de cette méthode sont les suivantes :

- Méthode basée sur RSA et ECC : Les deux méthodes offrent une sécurité plus élevée dans la communication vidéo. Le RSA est utilisé pour chiffrer l'ID vidéo et l'ECC pour générer des morceaux de vidéo chiffrés.

- Technique à clés multiples : Des clés distinctes sont générées pour chaque fragment de données vidéo et une clé distincte pour les métadonnées vidéo.

- Génération automatique de clés : Une méthode de génération de clés basée sur des équations est mise en œuvre pour réaliser une génération de clés dynamique et automatique. Cette fonction permet à l'algorithme de générer une clé unique pour chaque flux vidéo.

Revue de la littérature

U. Zia et al. [20] ont proposé un générateur de nombres pseudo-aléatoires basé sur la théorie du chaos capable de générer un nombre unique et indépendant qui peut être utilisé dans les techniques de cryptographie. Cette méthode permet la génération automatique et adaptative de nombres aléatoires. Le procédé utilise une structure en treillis pour relier le nœud voisin via un facteur de couplage. Ici, une famille de cartes chaotiques symétriques a été considérée comme les cartes locales pour les treillis de cartes couplées (CML) et une approche adaptative est proposée pour la sélection des paramètres de contrôle. Les paramètres de contrôle supplémentaires contribuent à la génération de séquences hautement chaotiques avec un grand espace clé pour les applications cryptographiques. Ils ont introduit le concept d'une carte symétrique généralisée

(GSM) comme carte locale pour le système CML. Ils ont également proposé le concept de valeurs adaptatives des paramètres de contrôle qui aboutit à un PRNG hautement chaotique et complexe.

H. Kezia et al. [21] ont développé un nouveau schéma de cryptage vidéo basé sur des cartes chaotiques. Ici, la séquence vidéo cryptée est prise, puis elle est divisée en cadres. Lorsque les images sont grandes, elles sont divisées en macro-blocs pour l'opération. Le système chaotique de Lorenz à haute dimension est employé pour confondre la position des pixels, et le concept de clé multiple est utilisé pour améliorer la sécurité du cryptosystème contre les attaques. Dans cette méthode, la séquence vidéo reçue est divisée en images, et chaque image aura une clé unique. Un ordre ascendant et descendant est appliqué à la séquence chaotique. Il est courant que les images vidéo soient plus grandes. Pour cette raison, les images vidéo sont divisées en blocs pour réduire le temps de calcul. Dans ce cas, la taille optimale des blocs est de (8×8). En fonction de la séquence triée, les positions des blocs sont modifiées. En utilisant ce processus, une image vidéo numérique est cryptée. Le décryptage est effectué dans le sens inverse. Il est symétrique au processus de cryptage et au fonctionnement du décryptage.

M. A. Khan et al. [22] ont proposé une technique d'authentification et de cryptage basée sur ECC pour les applications IoT. Dans ce travail, le coût de calcul et le retard dans le traitement des données médicales détectées ont été analysés et ont démontré le traitement rapide de l'ECC. Ici, un cadre en couches a été proposé pour le système de santé. La sécurité des données a été réalisée en utilisant l'algorithme ECC. L'étude démontre la force de l'ECC par rapport au RSA. L'étude affirme qu'une clé RSA de 1024 bits est équivalente à une clé ECC de 160 bits. L'ECC présente donc plus d'avantages que le RSA lorsque l'application génère des clés multiples pour le cryptage. Les données sont cryptées à l'aide de la technique de cryptage ECC amélioré (IECC).

L'ECC amélioré est basé sur une courbe avec un point de base spécifique dérivé de fonctions de nombres premiers.

R. Imam et al. [23] ont passé en revue les techniques cryptographiques basées sur RSA et ont suggéré l'adéquation des techniques cryptographiques pour diverses applications. Cette revue discute des différentes implémentations et améliorations de RSA dans de multiples domaines. La littérature relative aux méthodes à clés multiples a été listée dans cet article et indique l'importance des techniques à clés multiples dans les applications de streaming.

N. Sen et al. [24] ont étudié les performances des techniques de cryptographie basées sur ECC pour les données vidéo. L'ECC est plus performant que toute autre technique de cryptographie asymétrique en raison de la taille réduite de la clé et de la rapidité des opérations de cryptage et de décryptage. Une application de streaming vidéo en temps réel utilisant la cryptographie à courbe elliptique (ECC) est démontrée ici. Cette mise en œuvre chiffre les flux vidéo à l'aide de la clé publique du récepteur et les envoie au client, qui les déchiffre à l'aide de sa clé privée. Selon cette étude, la méthode de cryptage optimale doit être choisie avec soin afin de protéger les données multimédias tout en assurant la diffusion en temps réel des vidéos. Pendant la phase de mise en œuvre, 18 courbes ECC ont été testées et analysées, et les résultats ont été comparés à des flux non chiffrés et chiffrés par AES. Sur la base des résultats, certaines courbes ECC ont été suggérées pour le cryptage du flux vidéo en temps réel.

Dans [25], un modèle de chiffrement hybride a été proposé qui utilise les chiffrages AES et ECC. Ce modèle combine la robustesse et la simplicité de l'ECDLP et de l'AES. Afin de sécuriser une variété de données multimédia, le système les convertit en un format de texte codé en base64 avant de les crypter. Elles sont ensuite cryptées à l'aide d'AES, pour lequel des clés aléatoires sont générées. Pour extraire la clé sous forme de texte, un équivalent du code QR est

généré sous forme d'image, que le système utilise ensuite pour extraire la clé. Ainsi, les clés AES sont protégées par une couche de sécurité supplémentaire.

Dans [26], une approche cryptographique hybride qui utilise RSA ou AES avec ECC pour le cryptage vidéo a été étudiée, et les performances des techniques ont été mesurées. En termes d'exigences de stockage et de temps de traitement, le protocole d'accord de clé authentifiée basé sur l'identité proposé offre des performances significatives lors de l'utilisation de chiffrements par blocs ECC pour l'authentification des utilisateurs. Le schéma RSA pose des problèmes importants, notamment en ce qui concerne les exigences de stockage et les temps de traitement. En raison de la génération de nombres aléatoires, la différence en pourcentage des deux paramètres change légèrement. D'autres réductions des exigences de stockage seront obtenues en combinant ECC avec des techniques de compression.

Z. Chen et al. [27] ont mis en œuvre une méthode de cryptage d'images utilisant le hachage SHA-3, RSA et la détection par compression. Ce modèle hybride atteint une sécurité plus élevée en utilisant une séquence chaotique en même temps que les méthodes énumérées ci-dessus.

R. Hegde et al. [28] ont conçu une technique de cryptage utilisant ECC. L'originalité de cette méthode est qu'elle utilise plusieurs courbes elliptiques pour améliorer la robustesse des données. Les métadonnées sont séparées et cryptées, puis intégrées dans une vidéo à l'aide d'un codage matriciel modifié optimisé. Il est proposé d'utiliser SHA-3 et la détection compressive pour le cryptage asymétrique des images afin d'empêcher l'accès non autorisé aux images privées. Une matrice aléatoire est générée pour obtenir une image prétraitée, et une opération d'addition modulaire est effectuée entre celle-ci et l'image brute. Ensuite, les valeurs de hachage de l'image prétraitée sont calculées par l'algorithme de hachage sécurisé de troisième génération (SHA-3) et sont regroupées et additionnées pour obtenir trois clés en clair. En utilisant l'algorithme Rivest-

Shamir-Adleman (RSA), trois clés de texte chiffré peuvent être obtenues en conséquence. Ensuite, un nouveau modèle de transformation mathématique (MTM) est conçu pour transformer toutes les clés en valeurs initiales pour un système chaotique. Ensuite, le flux de clés est calculé en conséquence. Troisièmement, l'image brute est comprimée par détection compressive (CS), puis confondue par des séquences aléatoires. Ensuite, l'application de la transformation en ondelettes discrètes (DWT) à l'image confuse génère quatre composantes haute et basse fréquence. Des séquences chaotiques sont à nouveau utilisées pour confondre les composantes basse fréquence, puis toutes les composantes sont à nouveau recombinées en une matrice. Après cela, on effectue une DWT inverse (IDWT) pour obtenir une image de chiffrement moyenne (MCI). Enfin, une autre matrice aléatoire est générée par des séquences chaotiques, et l'image de chiffrement finale est obtenue par une opération d'addition modulaire à la MCI.

S. H. Murad et al. [29] ont étudié des modèles de cryptographie hybride à deux et trois niveaux appliqués à la sécurité du cloud. Dans l'approche à deux niveaux, quatre chiffrements symétriques ont été utilisés pour le chiffrement des données en masse : DES, 3DES, Blowfish et AES, ainsi que le chiffrement asymétrique RSA pour le chiffrement de la clé secrète. Tout d'abord, la clé symétrique secrète et la paire de clés RSA sont générées. Les fichiers de données sont ensuite chiffrés à l'aide de l'un ou l'autre des ciphers (DES, 3DES, Blowfish ou AES). Le chiffrement à clé secrète est effectué à l'aide de la clé publique et du chiffrement RSA. En utilisant la clé privée avec RSA, la clé secrète symétrique est récupérée à l'extrémité de réception. Une approche à trois niveaux utilise un double chiffrement en plus de RSA des quatre mêmes chiffrements symétriques. Les modèles à trois niveaux génèrent deux clés symétriques secrètes, puis une paire de clés RSA, et chiffrent ensuite deux fois les données d'entrée à l'aide de l'un ou l'autre des ciphers (DES, 3DES, Blowfish et AES). Dans la phase d'encapsulation des clés, la clé publique est utilisée pour chiffrer

les clés secrètes à l'aide du chiffrement RSA. Enfin, les serveurs en nuage reçoivent les données et les clés cryptées. Pour récupérer le fichier original, le récepteur doit d'abord récupérer les clés secrètes et décrypter les données chiffrées deux fois.

M. K. Ghosh et al. [30] ont proposé une méthode hybride pour assurer la confidentialité et la sécurité des données sur Internet. L'algorithme hybride proposé utilise RSA et Diffie-Hellman, où Diffie-Hellman agit comme un agent de transmission de clé sécurisée pour RSA et RSA assure la sécurité du message. Les clés Diffie-Hellman sont utilisées pour modifier et récupérer le texte chiffré original avant et après l'envoi. L'utilisateur peut choisir différentes tailles de clés RSA en fonction de la quantité de secret requise. Un seul algorithme hybride est utilisé pour le cryptage et le décryptage, c'est-à-dire qu'il relève d'un algorithme cryptographique à clé privée. Mais comme la taille de la clé dans l'algorithme cryptographique à clé privée est plus grande, la transmission consommerait une large bande passante.

J. Zhang et al. [31] ont proposé une méthode pour le codage des données vidéo en utilisant une méthode de codage en couches. Ici, la couche de base est chiffrée à l'aide de l'algorithme LEX. La clé de chiffrement de LEX est générée à l'aide de la technologie de chiffrement du chaos, et les données de la couche d'amélioration sont brouillées à l'aide d'un algorithme de permutation aléatoire basé sur le système chaotique.

L. Yu et al. [32] ont discuté des applications de l'algorithme de cryptage hybride dans les titres logiciels. Le travail discute des utilisations de la méthode hybride dans l'amélioration de la sécurité dans les logiciels de surveillance vidéo.

M. A. Khan et al. [33], afin d'accomplir la technique de cryptage hybride, les techniques de cryptage de données utilisant les séries de Fibonacci, la logique XOR, la séquence PN sont étudiées, analysées et leurs performances sont comparées dans ce travail. Le message est divisé en

trois parties et ces trois techniques différentes sont appliquées à ces parties et les performances sont à nouveau analysées. La base de ce travail est l'application de ces trois méthodes différentes à différentes parties du même message avec deux clés, à savoir la clé de segmentation et la clé de cryptage pour fournir une authentification et une validation supplémentaires.

C. L. Chowdhary et al. [34] ont proposé une analyse pour effectuer le cryptage et le décryptage d'images par hybridation de la cryptographie à courbes elliptiques (ECC) avec le chiffrement de Hill (HC), ECC avec la norme de cryptage avancée (AES) et ElGamal avec le chiffrement double Playfair (DPC). Avec les algorithmes hybrides, les algorithmes symétriques offrent vitesse et facilité de mise en œuvre, et les algorithmes asymétriques offrent une plus grande sécurité. La cryptographie à clé asymétrique est fournie par ECC et ElGamal, tandis que HC, AES et DPC fournissent une cryptographie à clé symétrique. En raison du temps nécessaire au cryptage et au décryptage, ECC et AES sont idéaux pour les communications à distance ou privées avec des images de petite taille. Selon la mesure métrique, ECC et HC fournissent une bonne solution globale pour le cryptage des images.

J. Dave et al. [35] ont examiné les avantages et les inconvénients du stockage des données biométriques dans le nuage. Pour les modèles biométriques unimodaux, l'article présente une méthode de cryptage hybride basée sur le cloud. Un algorithme générique a été proposé. En outre, l'impact global des algorithmes existants et les défis associés à leur utilisation sont brièvement discutés.

M. Hamdi et al. [36] ont proposé un algorithme de cryptage hybride basé sur les ciphers de bloc et de flux utilisant des systèmes chaotiques. Le schéma proposé adopte deux opérations principales : l'une pour générer un bloc de données pseudo-aléatoire qui sera utilisé pour le chiffrement par flux, et la seconde pour créer des tables de substitution et de permutation dans

l'étape initiale et effectuer des rondes pour les processus de confusion et de diffusion dans le chiffrement par blocs.

S. Chen et al. [37] ont conçu un système chaotique 6-D à domaine réel basé sur des principes fondamentaux d'anti-contrôle et ont ensuite développé l'algorithme Verilog HDL associé. Une plateforme matérielle FPGA équipée d'une puce XUP Virtex-II est utilisée pour mettre en œuvre un système de communication vidéo sécurisé basé sur le chaos en utilisant l'algorithme Verilog HDL proposé. En conséquence, des expériences matérielles sont menées pour démontrer le mécanisme de fonctionnement.

D. Das [38] a exploré un modèle hybride unique en exploitant la puissance de l'automatisation pour les tests de sécurité au niveau de l'acquisition/agrégation vidéo et en amalgamant les meilleures pratiques des tests de sécurité traditionnels effectués au niveau de l'application vidéo de l'utilisateur. Cette méthodologie hybride couvre toutes les phases de la chaîne de valeur du streaming vidéo, de l'origine à la lecture. Elle permet donc d'obtenir une couverture maximale des tests sur plusieurs appareils de lecture dans des conditions de charge de travail multiples. Dans cette méthodologie, on applique des conditions réelles, y compris la latence, le retard et tout ce qui a trait à la congestion.

J. K. Joshi et al. [40] ont proposé une méthode dans laquelle le fichier vidéo donné par l'utilisateur est divisé en blocs soumis au processus RA-AES, qui utilise une clé de 128 bits et le convertit dans un format illisible. Chaque caractère du bloc est ensuite transformé en une valeur ASCII décryptée du côté du récepteur. Le chiffrement à l'aide d'une clé plus longue est plus compliqué à casser que celui effectué à l'aide d'une clé plus petite. DES utilise une clé de 64 bits, 3DES utilise trois clés de 64 bits, tandis que RA-AES utilise plusieurs clés (128, 192, 256).

J. K. Liu et al. [41] ont proposé une infrastructure qui permet aux utilisateurs de partager et de rechercher des vidéos en temps réel sur une plateforme mobile. La plateforme permet à l'utilisateur d'identifier les récepteurs de sa connexion pour partager le contenu vidéo. Les récepteurs placés par les utilisateurs peuvent accéder au fichier, mais pas les autres. L'architecture proposée utilise une plateforme cloud pour partager les contenus et des techniques cryptographiques pour sécuriser les contenus.

E. Lin et al. [42] examinent les concepts et les approches de la gestion des droits numériques de la vidéo et décrivent les méthodes permettant d'assurer la sécurité, y compris les rôles du cryptage et du filigrane vidéo. Les efforts et les problèmes actuels sont décrits en matière de cryptage, de filigrane et de gestion des clés.

R. Ahuja et al. [43] ont développé une méthode pour protéger le droit d'auteur d'une vidéo numérique tout en l'encodant en utilisant la norme MPEG. Le même filigrane a été proposé pour être inséré à deux endroits différents dans les trames intra-codées. Par rapport aux trames bidirectionnelles (trames B) et aux trames de prédiction (trames P), ces trames ont une composante de luminance élevée, de sorte que même de légères altérations de ces trames ne dégraderont pas la perceptibilité de la technique de filigrane vidéo proposée. Dans le cadre du test de robustesse, des attaques intentionnelles et non intentionnelles ont été appliquées, ce qui a donné des résultats satisfaisants pour la plupart des attaques. Grâce au nombre suffisant de blocs de transformée en cosinus discrète quantifiés dans une seule image I, il est possible d'ajuster la taille des bits du filigrane dans une seule image. Cela permet de loger une grande quantité de filigranes dans l'objet vidéo.

L. Mou [44] a proposé une architecture pour identifier la propriété du contenu. Côté serveur, la propriété de chaque composant du contenu est identifiée et signalée comme un élément

de la description de la présentation du média. Côté client, la signalisation de la propriété est analysée à partir de la description de la présentation du média, et chaque composant du contenu est traité en conséquence lors de la présentation. Avant la diffusion en continu, la propriété de chaque composant de contenu multimédia est d'abord identifiée à l'aide du filigrane ou de l'empreinte digitale du média, puis signalée dans la description de la présentation du média avec une liste d'opérations possibles décidée au préalable par son propriétaire. Lors de la présentation, chaque composant du contenu sera traité en fonction des informations de signalisation. Des expériences sur de grands ensembles de données démontrent l'efficacité et l'efficience de la méthode proposée.

J. Ning et al [45] ont utilisé la technique de recherche croisée oblivious pour réduire la complexité de la recherche de PEKS via, qui a pris en charge des modèles de requête expressifs et une mise à jour flexible du champ de mots clés.

Il ressort de cette littérature que le modèle hybride est la méthode la plus appropriée pour améliorer la robustesse des données sécurisées. De plus, les techniques de cryptage basées sur l'ECC sont plus adaptées aux applications de streaming vidéo en temps réel, car l'ECC génère une petite clé et est rapide à traiter. Cependant, les méthodes asymétriques augmentent la complexité du décryptage ; c'est pourquoi un modèle hybride utilisant RSA et ECC est considéré dans ce travail.

Références

[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 Transactions on Broadcasting*, vol. 64, no. 1, pp. 11–25, 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, January 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," 4 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," 2021. [Online]. Available: https://dx.doi.org/10.21227/s7x7-wd58

[5] "Cisco Annual Internet Report - Cisco Annual Internet Report (2018–2023) White Paper." [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." [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," *Proceedings of the 7th Conference on Emerging Networking EXperiments and Technologies, CoNEXT'11*, 2011.

[8] X. Huang, D. Arnold, T. Fang, and J. Saniie, "A chaotic-based encryp- tion/decryption system for secure video transmission," in *2021 IEEE International Conference on Electro Information Technology (EIT)*, 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: Chal- lenges and perspectives," *EURASIP J. Inf. Secur.*, vol. 2008, no. 1, dec 2008.

[10] A. Murtaza, S. J. Hussain Pirzada, and L. Jianwei, "A new symmetric key encryption algorithm with higher performance," in *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2019, pp. 1–7.

[11] S. Kansal and M. Mittal, "Performance evaluation of various symmetric encryption algorithms," in *2014 International Conference on Parallel, Distributed and Grid Computing*, 2014, pp. 105–109.

[12] S. Kumar, M. S. Gaur, P. Sagar Sharma, and D. Munjal, "A novel approach of symmetric key cryptography," in *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, 2021, pp. 593–598.

[13] M. Dworkin, E. Barker, J. Nechvatal, J. Foti, L. Bassham, E. Roback, and J. Dray, "Advanced encryption standard (aes)," 2001-11-26 2001.

[14] Y. Shen, Z. Sun, and T. Zhou, "Survey on asymmetric cryptography algorithms," in *2021 International Conference on Electronic Information Engineering and Computer Science (EIECS)*, 2021, pp. 464–469.

[15] S. Kumar, B. K. Singh, Akshita, S. Pundir, S. Batra, and R. Joshi, "A survey on symmetric and asymmetric key based image encryption," in *2nd International Conference on Data, Engineering and Applications (IDEA)*, 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, p. 120–126, feb 1978. [Online]. Available: https://doi.org/10.1145/359340.359342

[17] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, Jan. 1987.

[18] A. J. Menezes and S. A. Vanstone, "Elliptic curve cryptosystems and their implementation," *Journal of Cryptology 1993 6:4*, vol. 6, pp. 209–224, 9 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 and Cryptography 2000 19:2*, vol. 19, pp. 173–193, 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," *SN Applied Sciences*, vol. 4, pp. 1–17, 2 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 *2008 16th International Conference on Advanced Computing and Communications*, 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. 52 018–52 027, 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. 155 949–155 976, 2021.

[24] N. Sen, R. Dantu, J. Vempati, and M. Thompson, "Performance analysis of elliptic curves for real-time video encryption," in *2018 National Cyber Summit (NCS)*, 2018, pp. 64–71.

[25] S. C. Iyer, R. Sedamkar, and S. Gupta, "A novel idea on multimedia encryption using hybrid crypto approach," *Procedia Computer Science*, vol. 79, pp. 293–298, 2016, proceedings of International Conference on Communication, Computing and Virtualization (ICCCV) 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 *2012 International Conference on Computing, Communication and Applications*, 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, p. 169676, 2022. [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," *Computer Standards & Interfaces*, vol. 48, pp. 173–182, 11 2016.

[29] S. H. Murad and K. H. Rahouma, "Implementation and performance analysis of hybrid cryptographic schemes applied in cloud computing environment," *Procedia Computer Science*, vol. 194, pp. 165– 172, 2021, 18th International Learning & Technology Conference 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 *2020 International Conference on Computer Science, Engineering and Applica- tions (ICCSEA)*, 2020, pp. 1–5.

[31] J. Zhang and X. Gao, "A hybrid encryption scheme for scalable video cod- ing based on h.264," in *2013 5th International Conference on Intelligent Networking and Collaborative Systems*, 2013, pp. 708–711.

[32] L. Yu, Z. Wang, and W. Wang, “The application of hybrid encryption algorithm in software security,” in *2012 Fourth International Conference on Computational Intelligence and Communication Networks*, 2012, pp. 762–765.
[33] M. A. Khan, K. K. Mishra, N. Santhi, and J. Jayakumari, “A new hybrid technique for data encryption,” in *2015 Global Conference on Communi- cation Technologies (GCCT)*, 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, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/18/5162
[35] J. Dave and M. Gayathri, “Hybrid encryption algorithm for storing uni- modal biometric templates in cloud,” in *Inventive Communication and Computational Technologies*, G. Ranganathan, X. Fernando, and F. Shi, Eds. Singapore: Springer Nature Singapore, 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, p. 1847–1858, feb 2021. [Online]. Available: https://doi.org/10.1007/s00500-020-05258-z
[37] S. Chen, S. Yu, J. Lü, G. Chen, and J. He, “Design and fpga-based realization of a chaotic secure video communication system,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2359– 2371, 2018.
[38] P. Yu, N. Zhang, S. Zhang, and Q. Wang, “Security mechanism of video content integrated broadcast control platform under triple play,” in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 2017, pp. 1–5.
[39] D. Das, “Automated security testing framework for validating content rights on video streaming devices,” in *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, 2019, pp. 516–521.
[40] J. K. Joshi, D. S. Bais, and A. N. Dubey, “An efficient and secure method for quality video streaming in mobile ad-hoc network,” in *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, 2015, pp. 75–79.
[41] J. K. Liu, M. H. Au, W. Susilo, K. Liang, R. Lu, and B. Srinivasan, “Secure sharing and searching for real-time video data in mobile cloud,” *IEEE Network*, vol. 29, no. 2, pp. 46–50, 2015.
[42] E. Lin, A. Eskicioglu, R. Lagendijk, and E. Delp, “Advances in digital video content protection,” *Proceedings of the IEEE*, vol. 93, no. 1, pp. 171–183, 2005.
[43] R. Ahuja, A. Kaur, V. Lamba, V. Kukreja, A. Agarwal, and M. Sharma, “Securing copyright of digital video by exploiting quantized dc coef- ficients,” in *2021 6th International Conference on Signal Processing, Computing and Control (ISPCC)*, 2021, pp. 423–427.
[44] L. Mou, “Ownership identification and signaling of multimedia content components,” in *2018 IEEE Conference on Multimedia Information Pro- cessing and Retrieval (MIPR)*, 2018, pp. 212–213.
[45] J. Ning, J. Chen, K. Liang, J. K. Liu, C. Su, and Q. Wu, “Efficient encrypted data search with expressive queries and flexible update,” *IEEE Transactions on Services Computing*, vol. 15, no. 3, pp. 1619–1633, 2022.

Chapitre 3 : Méthode proposée

Les technologies prises en compte dans ce travail sont présentées dans ce chapitre. Pour créer le modèle hybride, la méthode proposée utilise les techniques cryptographiques RSA, ECC et AES. De plus, comme le modèle proposé vise principalement à améliorer la sécurité dans les applications de communication vidéo, le chapitre aborde ses défis et ses problèmes. Enfin, la méthode proposée a fait l'objet d'une discussion approfondie.

Comme nous l'avons vu dans le chapitre précédent, les applications de streaming vidéo utilisent différentes méthodes pour sécuriser le contenu vidéo sur une plateforme de streaming. Méthodes de sécurisation du streaming vidéo :

- Techniques de cryptographie,
- Livraison HTTPS,
- Paywall crypté SSL/TLS,
- Vidéo protégée par mot de passe,
- Restrictions géographiques (IP),
- Restrictions sur les référents
- Centre de données vidéo et CDN sécurisés.

Dans ce chapitre, les techniques de cryptographie ont été prises en compte et développées. Le chapitre est divisé en plusieurs sections : Technique de cryptographie, AES, RSA, ECC, défis et problèmes liés à la sécurité des communications vidéo, et méthode proposée.

Technique de cryptographie

La cryptographie est une méthode de dissimulation des données sur un canal de communication. Il faut des compétences pour dissimuler des données à des inconnus. À mesure

que la technologie progresse, le besoin de sécurité des données sur les canaux de communication devient de plus en plus important. La cryptographie est utilisée pour sécuriser les connaissances.

La cryptographie se divise en trois catégories : les systèmes symétriques, les systèmes asymétriques et les protocoles cryptographiques. Les algorithmes symétriques et asymétriques peuvent être utilisés pour sécuriser les communications Internet. Les protocoles cryptographiques concernent la mise en œuvre des algorithmes cryptographiques. Chaque navigateur web utilise le schéma TLS (Transport Layer Security), qui est un exemple de protocole cryptographique.

La cryptographie à clé symétrique (également appelée cryptographie à clé secrète) utilise une seule clé pour le cryptage et le décryptage. Tout message crypté avec la clé publique ne peut être décodé qu'avec la même clé publique.

- Utiliser la même clé pour crypter et décrypter un message.
- Les algorithmes de cryptage et de décryptage sont distincts.

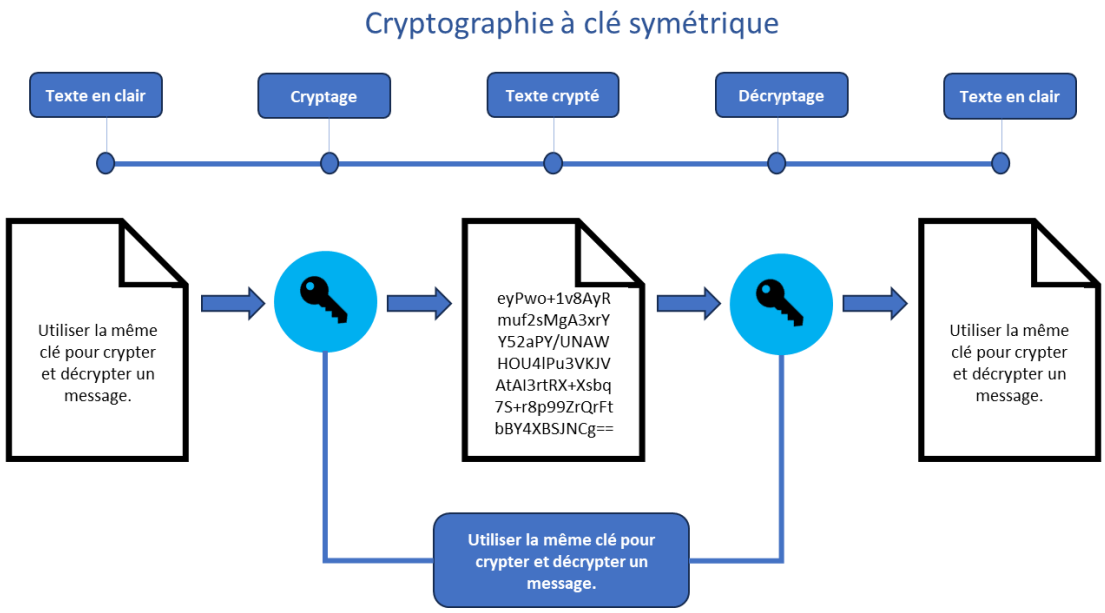


Figure 1 Chiffrement à clé symétrique

Le cryptage à clé asymétrique (également appelé cryptage à clé publique) utilise deux clés distinctes pour le cryptage et le décodage des messages. La clé publique est mise à la disposition du public et peut être utilisée pour chiffrer des messages. La clé privée reste privée et peut être utilisée pour décoder les messages reçus. L'algorithme RSA est une technique de cryptage à clé asymétrique.

- Il permet de crypter une communication à l'aide d'une clé (clé publique).
- Une autre (clé privée) est nécessaire pour décrypter un message.

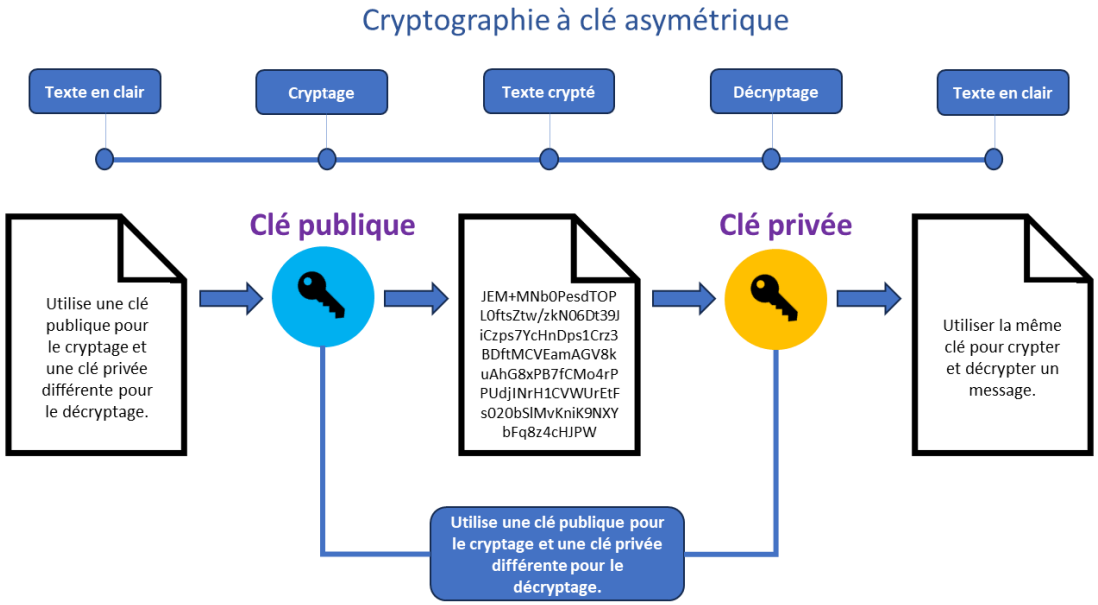


Figure 2 Chiffrement à clé asymétrique

Norme de cryptage avancée (AES)

L'Advanced Encryption Standard est l'algorithme de chiffrement symétrique le plus populaire et le plus utilisé que l'on puisse rencontrer de nos jours (AES). Il a été découvert qu'il est au moins six fois plus rapide que le triple DES. La taille de la clé du DES étant trop petite, un remplacement était nécessaire. On pensait qu'il était vulnérable à un assaut de recherche exhaustive

des clés lorsque la puissance de traitement augmentait. Le triple DES était censé pallier cet inconvénient, mais on a découvert qu'il était lent.

Les caractéristiques de l'AES sont les suivantes :

- Données de 128 bits, clés de 128/192/256 bits
- Chiffrement par blocs à clé symétrique
- Triple-DES est plus puissant et plus rapide
- Logiciel écrit en C et Java

AES est un chiffrement itératif, par opposition au chiffrement de Feistel. Il est construit sur le "réseau de substitution-permutation". Il est constitué d'une séquence d'opérations liées entre elles, dont certaines consistent à remplacer des entrées par certaines sorties (substitutions) et d'autres à déplacer des bits (permutations).

Étonnamment, AES base tous ses calculs sur des octets plutôt que sur des bits. Par conséquent, AES considère les 128 bits d'un bloc de texte en clair comme 16 octets. Ces 16 octets sont organisés en quatre colonnes et quatre lignes pour le traitement matriciel.

Contrairement au DES, le nombre de tours dans AES est variable et est déterminé par la longueur de la clé. AES utilise dix tours pour les clés de 128 bits, douze tours pour les clés de 192 bits et quatorze tours pour les clés de 256 bits. Chacun de ces tours utilise une clé unique de 128 bits dérivée de la clé AES originale.

Le schéma de la structure d'AES est donné dans la Figure 3.

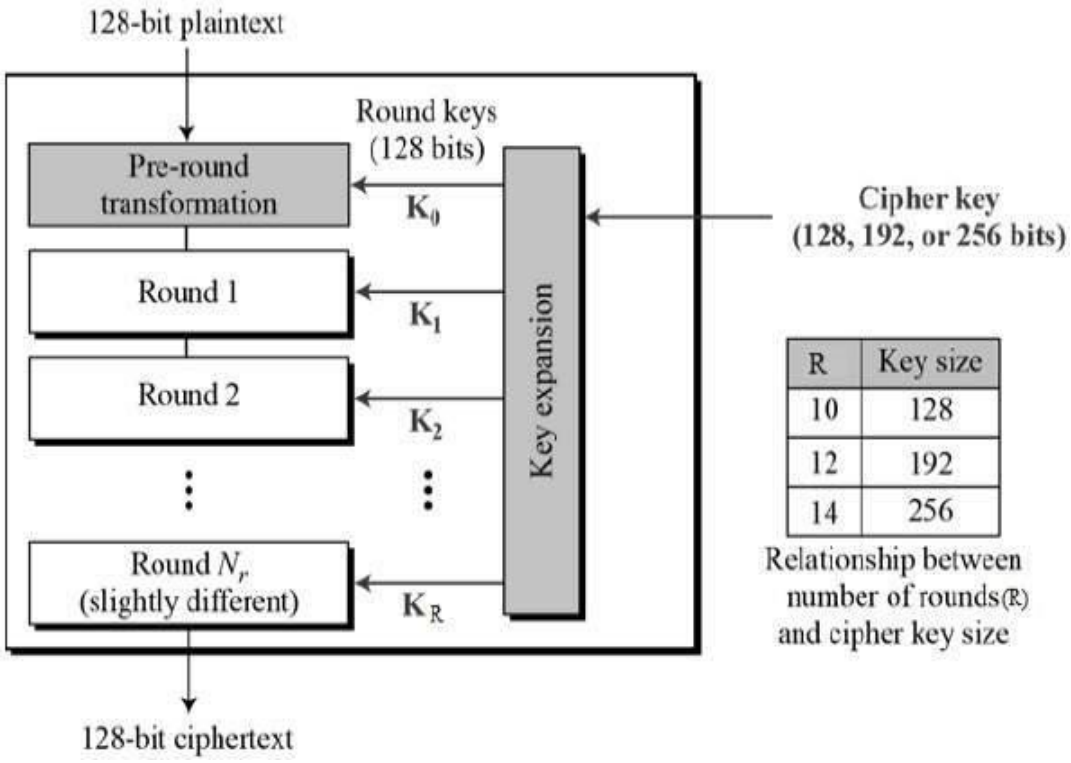


Figure 3 Structure de l'AES

Fonctionnement interne d'un tour

L'algorithme commence par une étape d'ajout de clé ronde suivie de 9 rondes de quatre étapes et d'une dixième ronde de trois étapes. Cela s'applique à la fois au cryptage et au décryptage, à l'exception du fait que chaque étape de l'algorithme de décryptage est l'inverse de son homologue dans l'algorithme de cryptage. Les quatre étapes sont les suivantes :

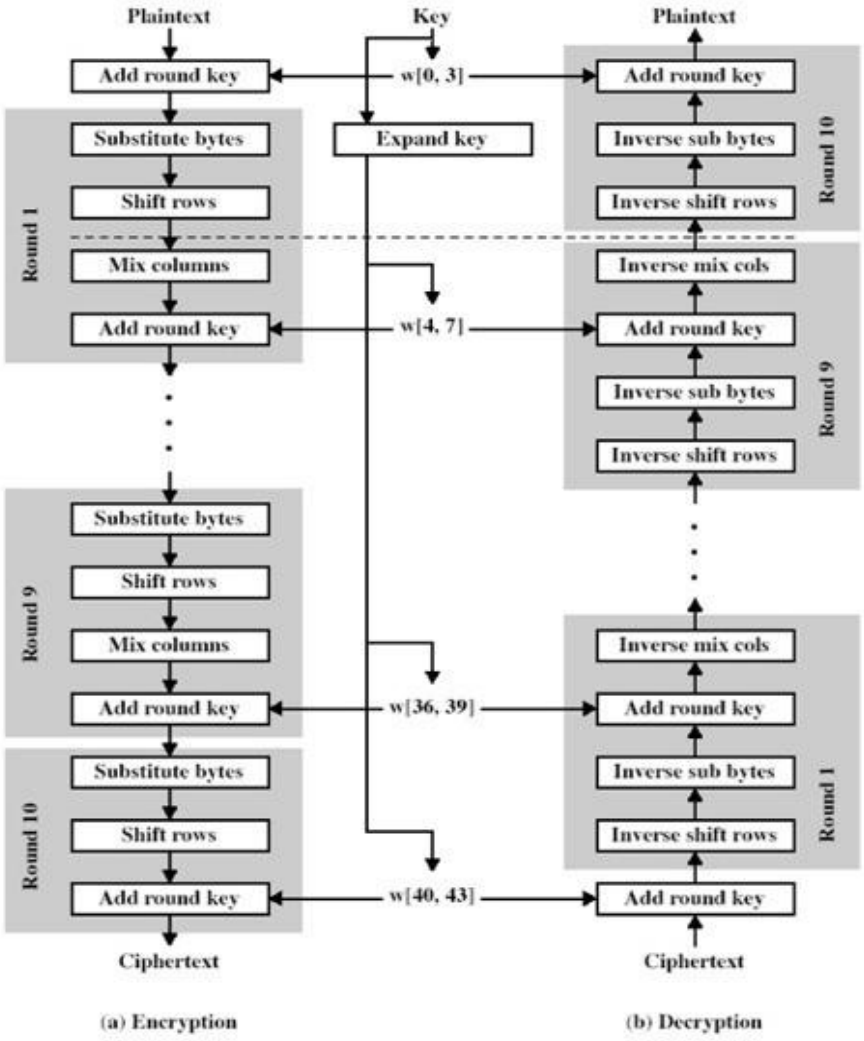


Figure 4 Structure globale de l'algorithme AES

- Substituer des octets
- Décaler des lignes
- Mélanger les colonnes
- Ajout de la clé du tour

Le dixième tour laisse simplement de côté l'étape de mélange des colonnes. Les neuf premiers tours de l'algorithme de décryptage sont les suivants :

- Lignes de décalage inversées

- Substitution inversée d'octets
- Clé d'ajout de tour inversée
- Mélange inversé de colonnes

Là encore, le dixième tour laisse simplement de côté l'étape des colonnes de mélange inversées. Nous allons maintenant examiner chacune de ces étapes plus en détail.

Octets de substitution

Cette étape (connue sous le nom de SubBytes) est simplement une consultation de table utilisant une matrice 16x16 de valeurs d'octets appelée s-box. Cette matrice est constituée de toutes les combinaisons possibles d'une séquence de 8 bits (28 = 16x16 = 256). Cependant, la s-box n'est pas une simple permutation aléatoire de ces valeurs et il existe une méthode bien définie pour créer les tables s-box. Les concepteurs de Rijndael ont montré comment cela a été fait, contrairement aux s-boxes de DES pour lesquelles aucune justification n'a été donnée. Nous ne nous préoccuperons pas trop ici de la façon dont les s-boxes sont constitués et nous pouvons simplement les considérer comme des tables de consultation.

Encore une fois, la matrice qui est exploitée tout au long du cryptage est connue sous le nom d'état. Nous nous intéresserons à la façon dont cette matrice est affectée à chaque tour. Pour ce cas particulier

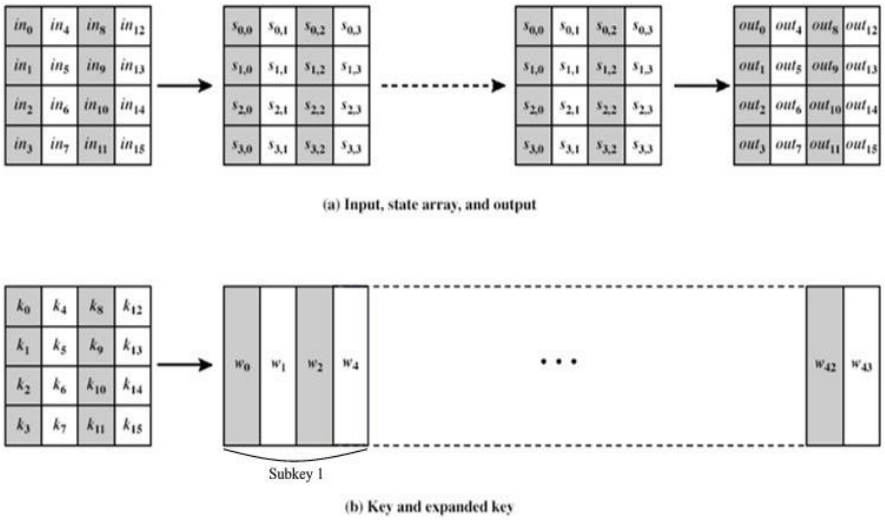


Figure 5 Structures de données dans l'algorithme AES

Chaque octet est mappé dans un nouvel octet de la manière suivante : le quartet le plus à gauche de l'octet est utilisé pour spécifier une ligne particulière de la boîte s et le quartet le plus à droite spécifie une colonne. Par exemple, l'octet {95} (les crochets représentent les valeurs hexagonales dans la norme FIPS PUB 197) sélectionne la ligne 9, colonne 5, qui contient la valeur {2A}. Cette valeur est ensuite utilisée pour mettre à jour la matrice d'état. La figure 6 illustre cette idée.

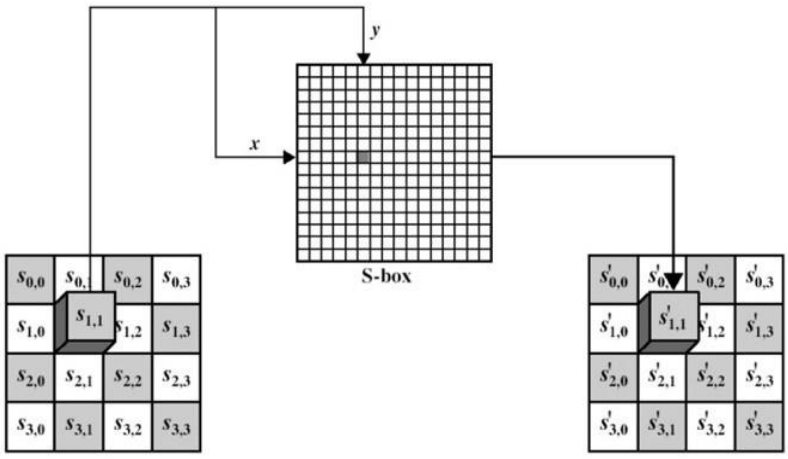


Figure 6 Étape de substitution d'octets de l'algorithme AES

La transformation inverse des octets de substitution (connue sous le nom de InvSubBytes) utilise une boîte ‘s’ inverse. Dans ce cas, ce que l'on souhaite, c'est sélectionner la valeur {2A} et obtenir la valeur {95}. Le tableau 7.4 montre les deux s-boxes, et l'on peut vérifier que c'est bien le cas.

Tableau 1. Deux s-boxes

(a) S-box

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

(b) S-box inverse

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

La s-box est conçue pour être résistante aux attaques cryptographiques connues. Plus précisément, les développeurs de Rijndael ont cherché une conception qui présente une faible corrélation entre les bits d'entrée et les bits de sortie, et la propriété que la sortie ne peut pas être

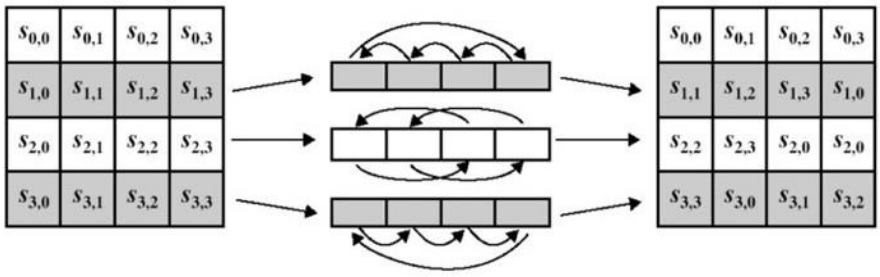
décrite comme une simple fonction mathématique de l'entrée. En outre, la s-box n'a pas de points fixes (s-box(a) = a) et pas de points fixes opposés (s-box(a) = ‾a) où ‾a est le complément binaire de a. La s-box doit être inversible si le décryptage est possible (Is-box[s-box(a)]= a), mais elle ne doit pas être son auto-inverse, c'est-à-dire s-box(a) /= Is-box(a).

Transformation Shift Row

Cette étape (connue sous le nom de ShiftRows) est illustrée à la figure 8. Il s'agit d'une simple permutation et rien de plus. Elle fonctionne comme suit :

- La première ligne d'état n'est pas modifiée.
- La deuxième rangée est décalée d'un octet vers la gauche de manière circulaire.
- La troisième ligne est décalée de 2 octets vers la gauche de manière circulaire.
- La quatrième ligne est décalée de 3 octets vers la gauche de manière circulaire.

Figure 7: Étape ShiftRows



La transformation Inverse Shift Rows (connue sous le nom de InvShiftRows) effectue ces décalages circulaires dans la direction opposée pour chacune des trois dernières rangées (la première rangée n'a pas été modifiée au départ).

Cette opération ne semble pas faire grand-chose, mais si vous réfléchissez à la façon dont les octets sont ordonnés dans l'état, vous constaterez qu'elle a un impact bien plus important. Rappelez-vous que l'état est traité comme un tableau de colonnes de quatre octets, c'est-à-dire que

la première colonne représente les octets 1, 2, 3 et 4. Un décalage d'un octet correspond donc à une distance linéaire de quatre octets. La transformation fait également en sorte que les quatre octets d'une colonne soient répartis sur quatre colonnes différentes.

Transformation colonne de mélange (Mix Column)

Cette étape (connue sous le nom de MixColumn) est fondamentalement une substitution mais elle utilise l'arithmétique de GF(28). Chaque colonne est traitée individuellement. Chaque octet d'une colonne est transformé en une nouvelle valeur qui est une fonction des quatre octets de la colonne. La transformation peut être déterminée par la multiplication matricielle suivante sur l'état (voir figure 9) :

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix} \quad (1)$$

Chaque élément de la matrice produit est la somme des produits des éléments d'une ligne et d'une colonne. Dans ce cas, les additions et multiplications individuelles sont effectuées dans GF (28). La transformation MixColumns d'une seule colonne j ($0 \leq j \leq 3$) d'état peut être exprimée comme :

sJ0,j = (2 - s0,j) ⊕ (3 - s1,j) ⊕ s2,j ⊕ s3,j.

sJ1,j = s0,j ⊕ (2 - s1,j) ⊕ (3 - s2,j) ⊕ s3,j

sJ2,j = s0,j ⊕ s1,j ⊕ (2 - s2,j) ⊕ (3 - s3,j)

sJ3,j = (3 - s0,j) ⊕ s1,j ⊕ s2,j ⊕ (2 - s3,j) (2)

où - désigne la multiplication sur le corps fini GF(28).

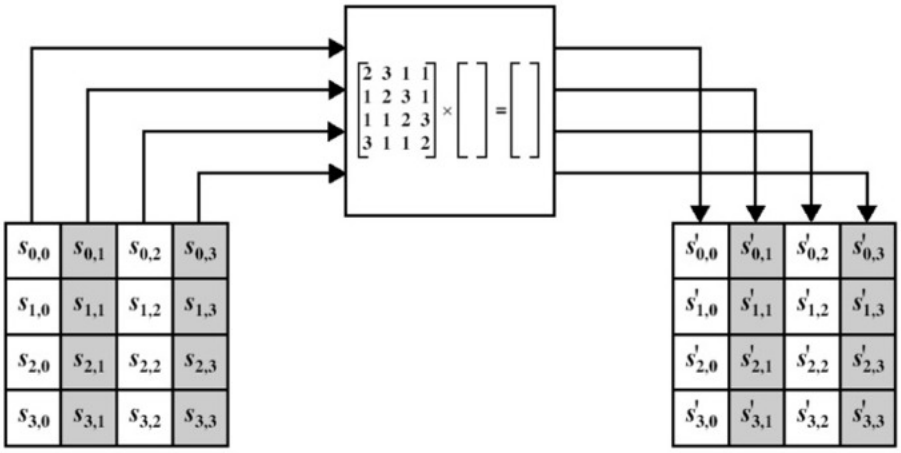


Figure 8 Étape MixColumns

À titre d'exemple, prenons la première colonne d'une matrice comme étant s0,0 = {87}, s1,0 = {6E}, s2,0 = {46}, s3,0 = {A6}. Cela signifie que s0,0 = {87} est mis en correspondance avec la valeur sj0,0 = 47, ce qui peut être vu en calculant la première ligne de l'équation 2 avec j = 0. Par conséquent, nous avons :

(02 - 87) ⊕ (03 - 6E) ⊕ 46 ⊕ A6 = 47

Donc, pour montrer que c'est le cas, nous pouvons représenter chaque nombre hexagonal par un polynôme :

{02} = x

{87} = x7 + x2 + x + 1

Multiplions ces deux-là ensemble et nous obtenons :

x - (x7 + x2 + x + 1) = x8 + x3 + x2 + x

Le degré de ce résultat étant supérieur à 7, nous devons le réduire modulo un polynôme irréductible m(x). Les concepteurs de AES ont choisi m(x) = x8 + x4 + x3 + x + 1. Ainsi, on peut voir que :

(x8 + x3 + x2 + x) mod (x8 + x4 + x3 + x + 1) = x4 + x2 + 1

Ceci est égal à 0001 0101 en binaire. Cette méthode peut être utilisée pour calculer les autres termes. Le résultat est donc

0001 0101

1011 0010

0100 0110

$\oplus$ 1010 0110

0100 0111 = {47}

Il existe en fait une façon plus simple de faire une multiplication modulo m(x). Si nous multiplions par {02}, tout ce que nous avons à faire est un décalage de 1 bit vers la gauche suivi

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

d'un XOR conditionnel par bit avec (00011011) si le bit le plus à gauche de la valeur originale (avant le décalage) était 1. La multiplication par d'autres nombres peut être considérée comme une application répétée de cette méthode. Stallings explique plus en détail pourquoi cela fonctionne, mais nous ne nous y intéresserons pas trop ici. Ce qu'il est important de noter cependant, c'est qu'une opération de multiplication a été réduite à un décalage et à une opération XOR. C'est l'une des raisons pour lesquelles AES est un algorithme très efficace à mettre en œuvre.

Le InvMixColumns est défini par la multiplication matricielle suivante :

(3)

On peut montrer que la première matrice de l'équation 1 est l'inverse de la première matrice de l'équation 3. Si nous étiquetons respectivement A et A-1 et si nous étiquetons l'état avant l'opération de mélange de colonnes comme S et après comme SJ, nous pouvons voir que :

$AS = Sj$

donc

$A^{-1}Sj$

$= A^{-1}AS = S$

Transformation de la clé du cycle d'ajout

Dans cette étape (appelée AddRoundKey), les 128 bits de l'état sont soumis à une opération XOR au sens du bit avec les 128 bits de la clé ronde. L'opération est considérée comme une opération par colonne entre les 4 octets d'une colonne d'état et un mot de la clé ronde. Cette transformation est aussi simple que possible, ce qui contribue à l'efficacité, mais elle affecte également chaque bit d'état.

Expansion de la clé AES

L'algorithme d'expansion de clé AES prend en entrée une clé de 4 mots et produit un tableau linéaire de 44 mots. Chaque tour utilise 4 de ces mots, comme le montre la figure 5. Chaque mot contient 32 octets, ce qui signifie que chaque sous-clé a une longueur de 128 bits. La figure 10 montre le pseudo-code pour générer la clé étendue à partir de la clé réelle.

```
KeyExpansion (byte key[16], word w[44])
{
   word temp
   for (i = 0; i < 4; i++)     w[i] = (key[4 * i], key[4 * i + 1], key[4 * i + 2], key[4 * i + 3]);
   for (i = 4; i < 44; i++)
   {
      temp = w[i − 1];
      if (i mod 4 = 0)     temp = SubWord (RotWord(temp)) ⊕ Rcon[i/4];
      w[i] = w[1 − 4] ⊕ temp;
   }
}
```

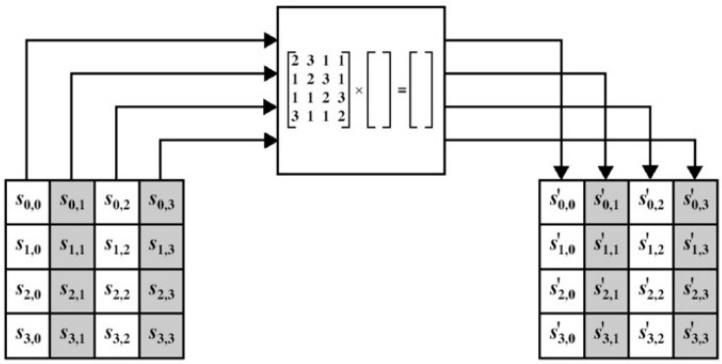


Figure 9 Pseudocode d'expansion de clé

La clé est copiée dans les quatre premiers mots de la clé étendue. Le reste de la clé étendue est rempli de quatre mots à la fois. Chaque mot ajouté w[i] dépend du mot immédiatement précédent, w[i-1], et du mot quatre positions en arrière w[i-4]. Dans trois cas sur quatre, un simple XOR est utilisé. Pour un mot dont la position dans le tableau w est un multiple de 4, une fonction

plus complexe est utilisée. La figure 11 illustre la génération des huit premiers mots de la clé étendue en utilisant le symbole g pour représenter cette fonction complexe. La fonction g se compose des sous-fonctions suivantes :

1. **RotWord** effectue un décalage circulaire vers la gauche d'un octet sur un mot. Cela signifie qu'un mot d'entrée [b0, b1, b2, b3] est transformé en [b1, b2, b3, b0].
2. **SubWord** effectue une substitution d'octet sur chaque octet de son mot d'entrée, en utilisant la boîte s décrite précédemment.

Le résultat des étapes 1 et 2 est soumis à une opération XOR avec la constante circulaire, Rcon[j].

La constante ronde est un mot dans lequel les trois octets les plus à droite sont toujours 0. Ainsi, l'effet d'un XOR d'un mot avec Rcon est d'effectuer un XOR uniquement sur l'octet le plus à gauche du mot. La constante de ronde est différente pour chaque ronde et est définie comme Rcon[j] = (RC[J], 0,0,0), avec RC[1]= 1, RC[j]= 2 RC[j-1] et avec une multiplication définie sur le champ GF(28).

L'expansion de clé a été conçue pour être résistante aux attaques cryptographiques connues. L'inclusion d'une constante de tour dépendante du tour élimine la symétrie, ou la similarité, entre la façon dont les clés de tour sont générées dans les différents tours.

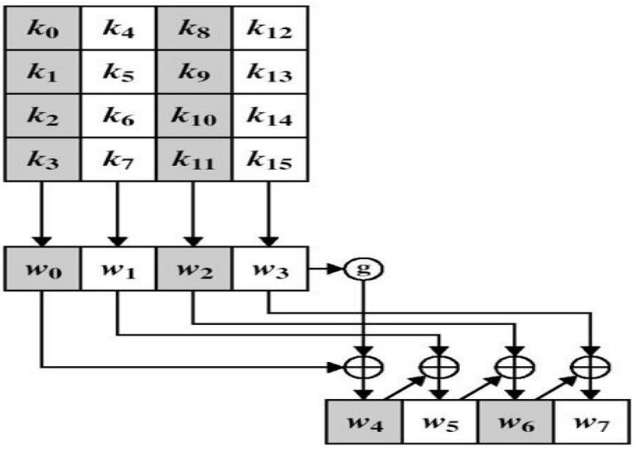


Figure 10 Expansion de clé AES

La figure 12 donne un résumé de chacun des tours. La colonne ShiftRows est représentée ici comme un décalage linéaire, ce qui donne une meilleure idée de la façon dont cette section contribue au chiffrement.

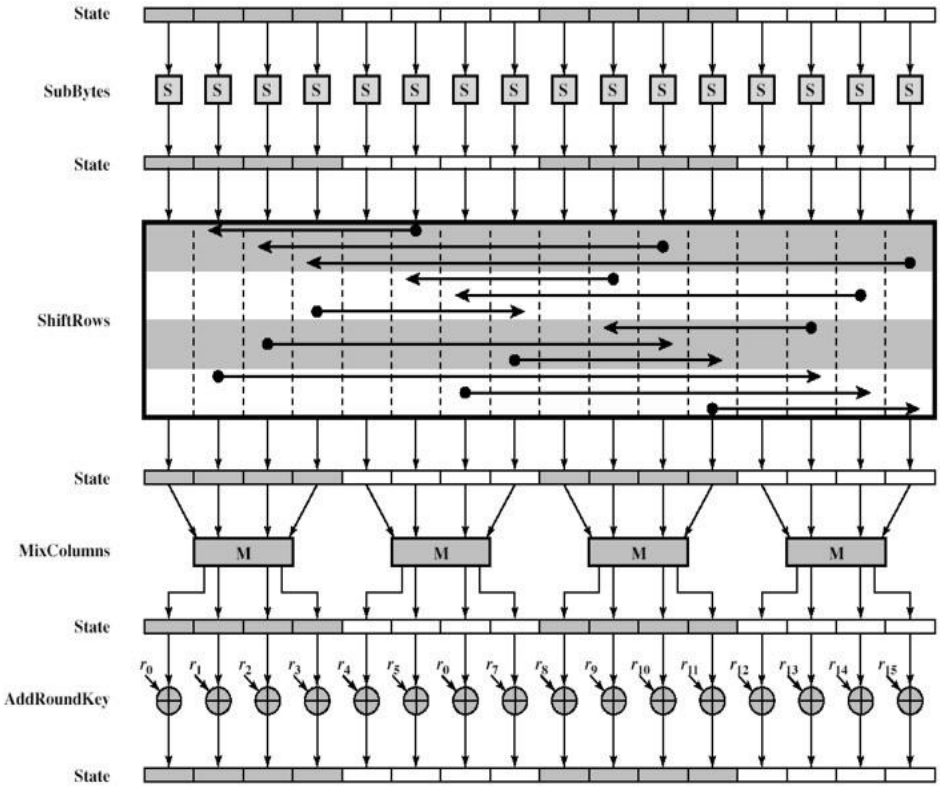


Figure 11 Cycle de cryptage AES

Chiffre inversé équivalent

Comme on peut le voir sur la figurecypher 3, les ciphers de décryptage ne sont pas identiques aux ciphers de cryptage. Cependant, la forme des programmes de clés est la même pour les deux. Cela présente l'inconvénient de nécessiter deux modules logiciels ou micrologiciels distincts pour les applications qui requièrent à la fois le cryptage et le décryptage. De plus, le décryptage est légèrement moins efficace à mettre en œuvre. Cependant, le cryptage a été jugé plus important que le décryptage pour deux raisons :

Pour le mode de chiffrement CFB et OFB (que nous avons vu auparavant mais que nous étudierons plus en détail par la suite), seul le chiffrement est utilisé.

Comme pour tout chiffrement par blocs, AES peut être utilisé pour construire un code d'authentification de message (qui sera décrit plus tard), et pour cela, seul le chiffrement est utilisé.

Cependant, si on le souhaite, il est possible de créer un chiffrement inverse équivalent. Cela signifie que le décryptage a la même structure que les algorithmes de cryptage. Cependant, pour y parvenir, un changement de programme de clés est nécessaire. Nous ne nous intéresserons pas à cette forme alternative, mais vous devez savoir qu'elle existe.

Cryptage par courbe elliptique

Une courbe elliptique est une équation cubique de la forme :

$y2 + axy + by = x3 + cx2 + dx + e$

où a, b, c, d et e sont des nombres réels.

Une opération d'addition particulière est définie sur les courbes elliptiques, et ce avec l'inclusion d'un point O, appelé point à l'infinité. Si trois points sont sur une ligne coupent une

courbe elliptique, leur somme est égale à ce point à l'infinité O (qui agit comme l'élément d'identité pour cette opération d'addition.

La figure 12 montre les courbes elliptiques y2 = x3 + 2x + 5 et y2 = x3 - 2x + 1.

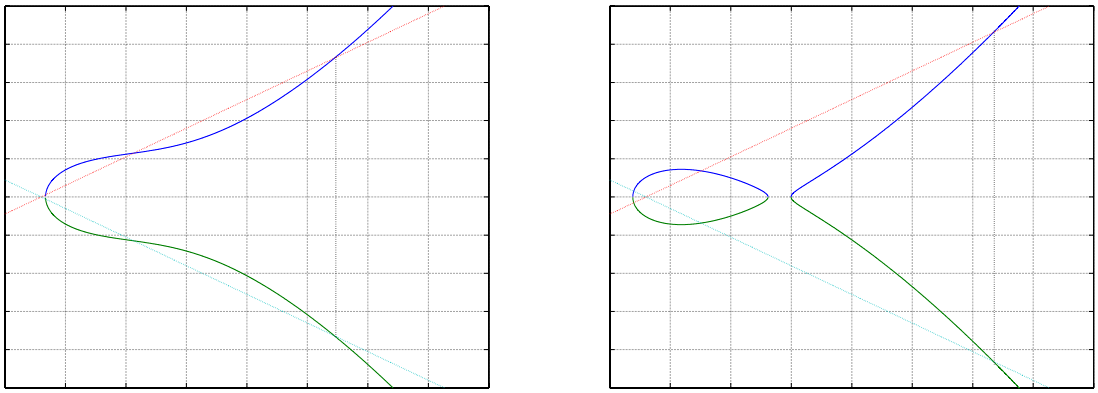


Figure 12: Courbes elliptiques y2 = x3 + 2x + 5 et y2 = x3 - 2x + 1

Courbes elliptiques sur les corps de Galois

Un groupe elliptique sur le champ de Galois Ep(a, b) est obtenu en calculant x3 + ax + b mod p pour 0≤ x < p. Les constantes a et b sont des entiers non négatifs inférieurs au nombre premier p et doivent satisfaire la condition :

$4a^3 + 27b^2 \bmod p \neq 0$.

Pour chaque valeur de x, on doit déterminer s'il s'agit ou non d'un résidu quadratique. Si c'est le cas, alors il y a deux valeurs dans le groupe elliptique. Si ce n'est pas le cas, alors le point n'est pas dans le groupe elliptique Ep(a, b).

Cryptage par courbe elliptique

La cryptographie par courbe elliptique peut être utilisée pour chiffrer des messages en clair, M, en messages chiffrés. Le message en clair M est codé en un point PM forment l'ensemble fini

de points du groupe elliptique, Ep(a, b). La first étape consiste à choisir un point générateur, G Ep(a, b), tel que la plus petite valeur de n telle que nG = O soit un très grand nombre premier. Le groupe elliptique Ep(a, b) et le point générateur G sont rendus publics.

Chaque utilisateur choisit une clé privée, nA < n et calcule la clé publique PA comme : PA = nAG. Pour chiffrer le point de message PM pour Bob (B), Alice (A) choisit un entier aléatoire k et calcule la paire de points PC en texte chiffré en utilisant la clé publique PB de Bob :

PC = [(kG), (PM + kPB)]

Après avoir reçu la paire de points en texte chiffré, PC, Bob multiplie le first point, (kG) avec sa clé privée, nB, puis ajoute le résultat au deuxième point de la paire de points en texte chiffré, (PM + kPB) :

(PM + kPB) - [nB(kG)] = (PM + knBG) - [nB(kG)] = PM

qui est le point de texte en clair, correspondant au message en clair M . Seul Bob, qui connaît la clé privée nB, peut supprimer nB(kG) du deuxième point de la paire de points de texte chiffré, c'est-à-dire (PM + kPB), et ainsi récupérer l'information en clair PM.

Sécurité de l'ECC

La force cryptographique du chiffrement par courbe elliptique réside dans la difficulté pour un cryptanalyste de déterminer le nombre aléatoire secret k à partir de kP et de P lui-même. La méthode la plus rapide pour résoudre ce problème (connu sous le nom de problème du logarithme de la courbe elliptique) est la méthode de factorisation ρ de Pollard.

La complexité de calcul pour casser le crypto système de la courbe elliptique, en utilisant la méthode Pollard ρ, est de $3{,}8 \times 10^{10}$ MIPS-années (c'est-à-dire des millions d'instructions par seconde multipliées par le nombre d'années requis) ou une taille de clé de la courbe elliptique de

seulement 150 bits. À titre de comparaison, la méthode la plus rapide pour casser RSA, en utilisant la méthode du crible général des champs de nombres pour factoriser l'entier composite n en deux nombres premiers p et q, nécessite 2x108 MIPS-années pour une clé RSA de 76^8 bits et $3x10^{11}$ MIPS-années avec une clé RSA de 1024 bits.

Si la longueur de la clé RSA est augmentée à 2048 bits, la méthode du crible général du champ de nombres aura besoin de $3x10^{20}$ MIPS-années pour factoriser n, tandis que l'augmentation de la longueur de la clé de la courbe elliptique à seulement 234 bits imposera une complexité de calcul de $1,6x10^{28}$ MIPS-années (toujours avec la méthode Pollard ρ).

Méthode proposée

Dans ce travail, nous avons proposé une nouvelle technique de cryptographie multi clé pour améliorer la sécurité de la communication vidéo. La méthode proposée utilise RSA et ECC comme bases pour réaliser la technique de cryptographie asymétrique.

Énoncé du problème

Dans une application de streaming vidéo, la vidéo est diffusée à la demande depuis le serveur média vers les appareils clients. La sécurisation du contenu vidéo numérique implique les mesures suivantes : accès conditionnel, authentification de l'utilisateur, contrôle de la copie du contenu et suivi du contenu vidéo. Ces mesures de sécurité sont généralement réalisées à l'aide de techniques de cryptographie. Cependant, la recherche d'une solution complète pour la sécurité de la vidéo numérique est un défi.

De nombreuses recherches ont été menées en cryptographie afin d'explorer les avantages des méthodes de cryptographie à clé asymétrique pour surmonter les problèmes de gestion des

clés. Les méthodes existantes ne prennent pas en charge les techniques dynamiques et automatiques à clés multiples qui permettraient de renforcer la sécurité des applications de communication vidéo. Par conséquent, une méthode de gestion des clés automatique et dynamique est nécessaire. Ce travail se concentre donc sur une technique de cryptage multi clé basée sur RSA et ECC.

Algorithmes proposés

La méthode proposée vise à générer des clés dynamiques multiples basées sur les techniques de cryptographie à clé asymétrique RSA et ECC. L'objectif des techniques proposées est le suivant :

1. Sécuriser les données vidéo sur la base du contenu et des identifications uniques du récepteur
2. Pour profiter des avantages des techniques de cryptage hybrides, la proposition utilise les techniques RSA, ECC et AES.
3. Amélioration de la sécurité par la technique de cryptage multi clé
4. Augmenter la sécurité avec les morceaux de vidéo
5. Réduire la gestion des clés multiples en utilisant des méthodes de génération automatique de clés

Les objectifs sont atteints en utilisant le modèle de génération de clés proposé, illustré à la figure 13.

Le processus est lancé en transmettant les données vidéo au module de génération de clés. Le module a besoin de la clé publique du récepteur.

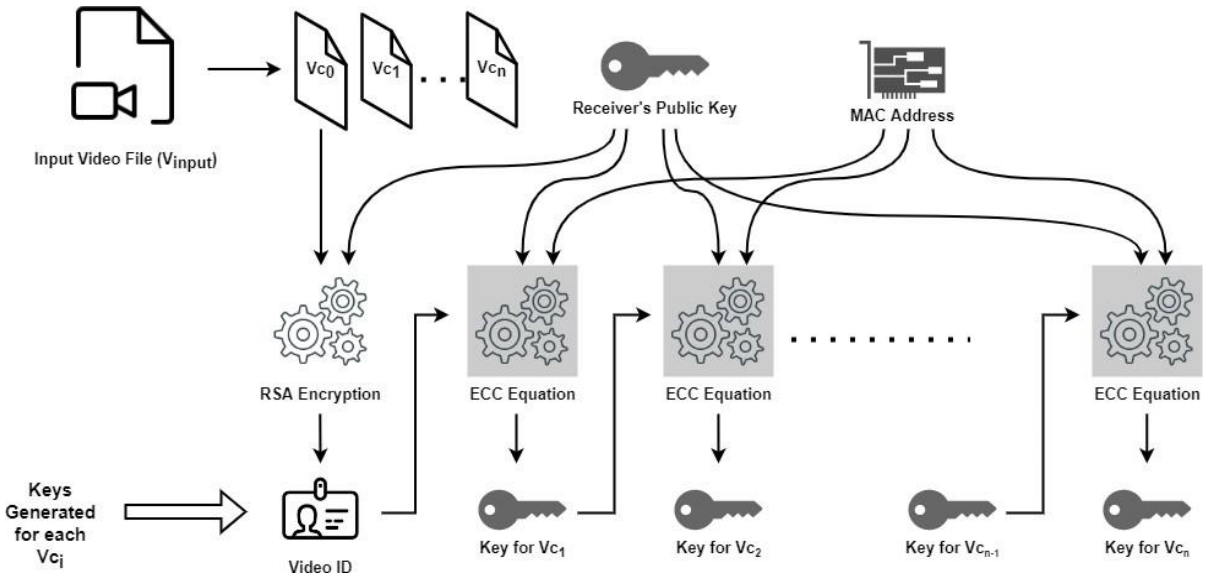


Figure 13 : Schéma de principe de la technique de génération de clés proposée

La clé R_P et l'adresse MAC R_{mac} du récepteur, qui améliorent l'unicité des clés générées au cours du processus. Ces attributs sont nécessaires tout au long de la communication vidéo ; ils sont donc stockés du côté de l'émetteur. Au départ, la vidéo est divisée en plusieurs morceaux d'une taille de 1 Mo. Ces morceaux de vidéo sont utilisés individuellement dans la génération de la clé.

L'algorithme 1 et la figure 14 décrivent les étapes de la génération de la clé. Le cryptage des morceaux de vidéo commence par la création de l'identification vidéo unique VID. Le VID est généré à l'aide du premier fragment vidéo V c0. Dans cette étape, le module récupère les 16 premiers octets de V c0 avant le cryptage. Ensuite, il le convertit en un format de chaîne de caractères base64. Plus tard, il est utilisé comme VID dans la génération de la clé ; par conséquent, le VID est stocké temporairement dans un fichier pour un accès rapide. Le premier morceau vidéo est crypté en utilisant le VID et la technique de cryptage RSA.

Algorithme 1 Flux de cryptage

1. Fichier vidéo d'entrée V_{input}
2. Générer des morceaux de vidéo V ci à partir de V_{input}
3. Récupérer la clé publique du récepteur de la clé R_P
4. Collecte de l'adresse MAC du récepteur R_{mac}
5. Générer le V_{ID} en utilisant le premier morceau de la vidéo
6. Stocker le V_{ID} dans un fichier temporaire
7. Crypter V_{c0} en utilisant RSA
8. Générer Key_a en utilisant l'équation : $x^3 + V_{ID}\, x + R_{mac}$
9. Crypter V_{c1} en utilisant Key_a et AES
10. **for** i:=2 n **do**
11. Générer Key_a en utilisant l'équation : x3 + Key_a x + R_{mac}
12. Cryptage de V_{ci} avec Key_a et AES
13. end for

La clé pour chaque mandrin vidéo est créée comme suit : le V_{ID} est utilisé pour générer la clé suivante. Ici, la méthode utilise la clé publique RP et l'adresse MAC R_{mac} du récepteur pour générer la clé de la deuxième partie de la vidéo. La méthode dérive la clé de l'équation ECC, c'est-à-dire,

$$y^2 = x^3 + ax + b \qquad (4)$$

La méthode proposée considère V_{ID} comme a et R_{mac} comme b dans l'équation ECC (Eq.4). Par conséquent, l'équation modifiée est la suivante :

$$Clé_a = x^3 + V_{ID} * x + R_{mac} \qquad (5)$$

L'équation 5 est utilisée uniquement pour le premier mandrin vidéo. Le mandrin vidéo utilisé est celui du deuxième mandrin ; il utilise Eq. 6. Ici, l'algorithme prend en compte la $Clé_a$ calculée précédemment au lieu du V_{ID}. Cette méthode est continue pour tous les autres mandrins de la vidéo.

$$Key_a = x^3 + Key_a * x + R_{mac} \quad (6)$$

La clé unique qui est générée pour chaque morceau de la vidéo est ensuite utilisée pour chiffrer le morceau de la vidéo à l'aide de l'algorithme AES. La méthode proposée ne partage aucune clé dans cette méthode multi clé et hybride. Comme décrit dans l'algorithme 1, la clé est générée à la volée pour la communication vidéo, ce qui permet d'obtenir une sécurité maximale. Une autre caractéristique essentielle de la méthode proposée est que même si un mandrin est compromis par des méthodes de force brute, le reste des données vidéo est sécurisé.

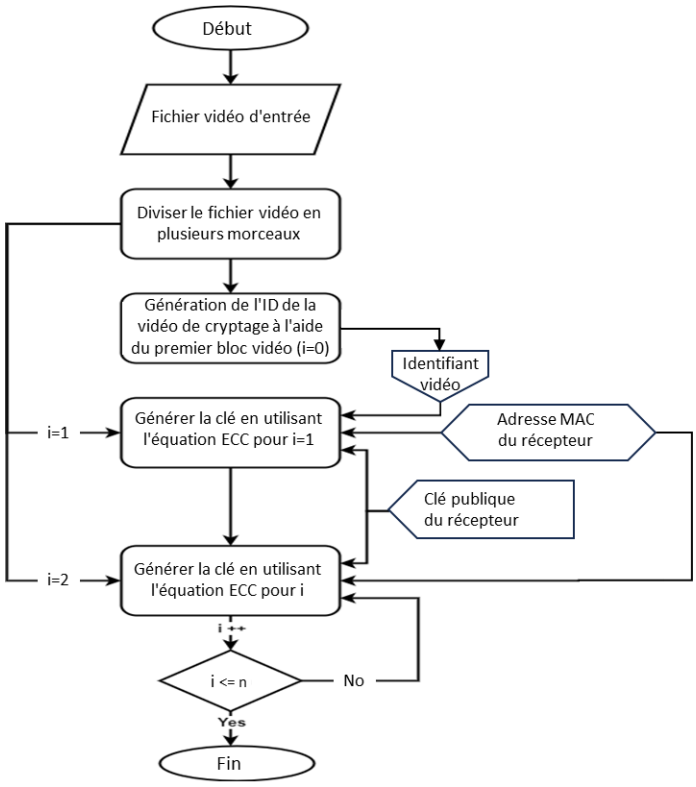


Figure 14 : Schéma de principe de la technique de génération de clé proposée

Chapitre 4 : Expérimentation et résultats

Les détails de l'implémentation et les résultats de l'expérimentation qui ont été obtenus ont été discutés dans ce chapitre. Les performances des méthodes suggérées sont évaluées à l'aide de diverses métriques dans l'analyse des résultats. Les détails de l'implémentation et de l'évaluation constituent les deux parties principales de ce chapitre.

Détails de l'implémentation

La plateforme mobile est utilisée pour mettre en œuvre la technique de cryptographie proposée. Ici, une application basée sur Android est utilisée pour mettre en œuvre les processus de cryptage et de décryptage ainsi que le streaming vidéo. Les éléments suivants font partie de la mise en œuvre :

- Module côté émetteur.
- Module côté récepteur.
- Base de données pour stocker les détails du récepteur.
- Module RSA
- Module d'équation ECC
- Module de traitement vidéo
- Module AES

Module côté émetteur

Aux fins de la diffusion en continu, les dispositifs émetteurs stockent le contenu vidéo et les métadonnées vidéo. Lorsqu'un destinataire demande à diffuser un fichier vidéo en continu, l'application de l'expéditeur démarre. Avant d'envoyer le contenu vidéo au destinataire,

l'expéditeur, un distributeur vidéo autorisé, vérifie l'identité du destinataire. Une base de données est utilisée pour conserver les informations sur le destinataire.

Les informations sur le destinataire sont recueillies au cours du processus d'inscription. où toutes les informations, y compris le nom d'utilisateur, le mot de passe, la clé publique et les informations sur le dispositif comme l'adresse MAC, sont récupérées et stockées dans la base de données de l'expéditeur.

Ce module comprend la mise en œuvre de la technique de cryptage proposée dans ce travail. Les fichiers vidéo sont d'abord divisés en parties par le module de traitement vidéo, puis chiffrés. Les algorithmes RSA et AES sont utilisés dans le processus de cryptage. Les parties vidéo sont cryptées avec la clé publique du récepteur et la clé privée de l'expéditeur. Le mandrin crypté est ensuite transmis en continu au récepteur.

La principale caractéristique de ce module est qu'il crypte le fichier vidéo en utilisant uniquement l'adresse MAC, le nom d'utilisateur et la clé publique du récepteur. Par conséquent, aucune clé n'est partagée avec le récepteur.

Module côté récepteur

Les principales tâches du récepteur sont le décryptage et la lecture de la vidéo. Le module reçoit des morceaux de vidéo chiffrés et les déchiffre à l'aide du module proposé. Les clés nécessaires sont obtenues à partir de la base de données du récepteur. Par conséquent, aucun échange de clés ne se produit dans cette méthode.

Le processus de décryptage utilise la clé publique de l'expéditeur et la clé privée du récepteur. Pour obtenir les premières parties de la vidéo, le module utilise une implémentation

RSA. Les données vidéo sont ensuite affichées sur l'unité d'affichage de l'appareil via l'application mise en œuvre. Il n'est donc pas nécessaire de sauvegarder les données vidéo reçues.

Le module ne décrypte la vidéo qu'en utilisant RSA afin de raccourcir le temps de traitement du premier morceau. Cela permet à l'écran de lire la vidéo immédiatement telle qu'elle a été reçue.

La principale caractéristique de ce module est que d'autres informations, notamment la clé privée du destinataire, l'adresse MAC et le nom d'utilisateur, peuvent être facilement obtenues alors que seule la clé publique de l'expéditeur est utilisée pour décrypter le fragment vidéo.

Mise en œuvre de la base de données

L'implémentation d'une base de données est rendue nécessaire par le besoin des applications de stocker les métadonnées vidéo et les informations du récepteur. Les informations suivantes sont conservées dans la base de données du côté de l'expéditeur : Informations sur le destinataire, y compris le nom d'utilisateur, l'adresse MAC du dispositif, la clé publique et d'autres informations spécifiques au compte. La clé publique de l'expéditeur et les métadonnées vidéo de la session de communication sont également stockées dans la base de données du côté du destinataire.

La base de données côté émetteur est maintenue à jour pendant toute la durée du fichier vidéo. Comme les informations de cette base de données sont statiques, elles sont toujours nécessaires. En revanche, les informations du récepteur sont mises à jour dès que les applications remarquent un changement.

Pour éviter toute violation de la sécurité par les utilisateurs, la base de données ne stocke pas la majorité des détails du côté du récepteur. Par conséquent, la base de données est légère et

simple, ce qui rend l'application plus portable et plus rapide. Seule la clé publique de l'expéditeur est nécessaire au module de décryptage pour recréer les données vidéo.

Module RSA

Les trois étapes de l'algorithme RSA sont la génération de la clé, le cryptage et le décryptage. Chaque clé générée sera associée à des paramètres spécifiques à l'algorithme, et le générateur produira une paire de clés publique et privée qui pourra être utilisée avec l'algorithme RSA. Pour générer les clés privées et publiques du récepteur dans cette implémentation, la méthode prend en compte les informations du récepteur. Les clés privées et publiques de l'expéditeur sont générées de manière similaire.

Ici, l'algorithme a été mis en œuvre sur la plateforme Android/Java, puis intégré à une application Android du côté du récepteur. Il est mis en œuvre du côté de l'expéditeur. Pour crypter et décrypter les mandrins vidéo, le module accède à la base de données et lit les informations de l'utilisateur.

Dans cette méthode proposée, seul le premier fragment vidéo est sécurisé par RSA. Le module chiffre la vidéo à l'aide d'octets partiels du morceau de vidéo et de la clé publique du récepteur, car le morceau de vidéo initial doit être traité rapidement. Les morceaux de vidéo suivants sont chiffrés avec AES, qui utilise une clé basée sur l'équation ECC dérivée de la clé publique et de l'adresse MAC du récepteur.

Module d'équation ECC

La clé dynamique est automatiquement générée par l'approche proposée à l'aide de l'équation ECC. L'ECC est connu pour son mécanisme de trappe, c'est pourquoi le module proposé

l'utilise. En plus de la clé précédemment générée qui a été couverte dans le chapitre précédent, le module prend également en compte la clé publique et l'adresse MAC du récepteur.

Dans cette mise en œuvre, la procédure utilise uniquement l'équation ECC pour obtenir la clé de chaque fragment vidéo à diffuser. La fonction de trappe améliore la sécurité des données vidéo en continu en évitant la reconstruction de la vidéo originale pour la redistribution. Le module utilise initialement un identifiant vidéo chiffré par RSA créé à partir de données vidéo partielles et de la clé publique du récepteur. La clé précédemment créée est ensuite utilisée comme l'une des entrées dynamiques par l'ECC pour récupérer la clé la plus récente pour le morceau.

Module de traitement vidéo

Le fichier vidéo est lu par le module côté expéditeur, qui crée également les blocs vidéo. Dans ce cas, les blocs vidéo ont été générés à l'aide du module FFMpeg. Une taille unitaire a été utilisée pour diviser la vidéo. Le module génère des morceaux à la trame la plus proche, quelle que soit la taille du morceau, que nous avons fixée à environ 1 Mo, mais qui peut être de n'importe quelle taille.

Métadonnées PEG standard a été utilisé pour diviser la vidéo en morceaux et lire les métadonnées du fichier vidéo. Cela rend la procédure plus évolutive et indépendante de la plateforme.

Module AES

À l'exception du fragment vidéo initial, tous les autres fragments vidéo sont cryptés et décryptés à l'aide de l'algorithme de cryptage AES. AES est implémenté par le module. Pour crypter les morceaux de vidéo, il prend en compte la clé créée par l'équation ECC. Le module est également utilisé pour décoder les morceaux de vidéo du côté du récepteur.

Résultats et discussion

La technique de cryptographie proposée a été évaluée dans cette section à l'aide de paramètres tels que le temps de génération de chaque clé, le temps de cryptage des fichiers avec des tailles de fichiers variables, le temps de cryptage des fichiers avec des tailles de morceaux variables, le nombre de clés générées et le délai de traitement de bout en bout entre la génération de la clé et le cryptage complet. Il en va de même pour le décryptage.

Délai pour diviser un fichier en morceaux

Le temps qu'il ait fallu à l'application pour créer les morceaux de vidéo à partir du fichier vidéo a été mesuré à l'aide de métrique du délai. L'une des principales contributions de ce travail est l'utilisation d'un chiffrement basé sur des morceaux de vidéo. Le temps de traitement impliqué dans le module de traitement vidéo est donc démontré par cette enquête. La taille du fragment est lue par l'utilitaire FFMpeg à partir de l'expéditeur. Le fragment est ensuite divisé en trame complètes les plus proches, ce qui donne une taille de fragment déterminée par l'utilisateur. Les résultats collectés ont été représentés graphiquement à l'aide de l'outil Excel.

La figure 15 illustre le temps nécessaire au module de traitement vidéo pour diviser le fichier vidéo en plusieurs morceaux. Le résultat montre que le délai augmente progressivement avec la taille du fichier, ce qui est simple à comprendre. Les résultats montrent que la technique n'allonge pas le temps de manière significatif. Cependant, le délai dans son ensemble est le résultat de la création des morceaux de du fichier vidéo. Cependant, cela est nécessaire pour obtenir une sécurité élevée.

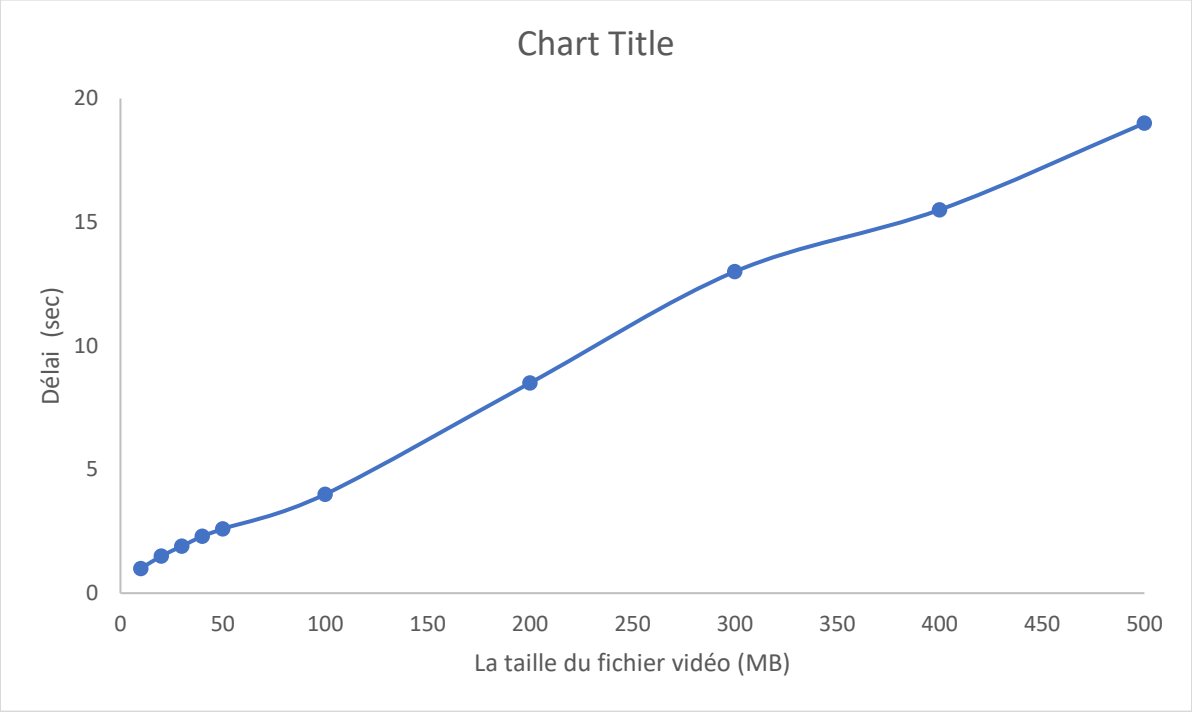


Figure 15 : Temps nécessaire au module de traitement vidéo pour diviser la vidéo en morceaux

Temps pour générer les clés

Pour déterminer l'impact de l'utilisation de la méthode à clés multiples dans le processus de cryptage et de décryptage, le temps nécessaire pour générer la clé a été calculé dans cette expérience. La méthode et l'équation utilisées pour dériver les clés sont les mêmes pour le cryptage et le décryptage. Ainsi, pour les besoins de l'analyse, le délai calculé pour le cryptage a été employé dans cette partie.

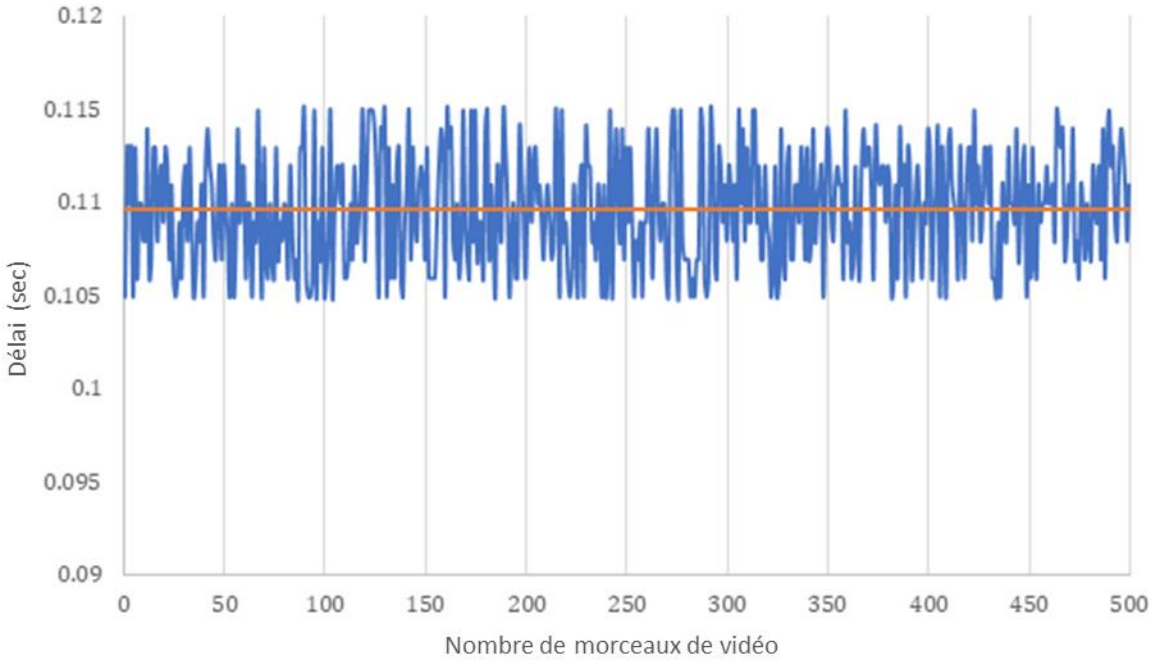


Figure 16 : Délai de génération de clés multiples

La figure 16 illustre le temps nécessaire à la génération de chaque clé. La clé publique du récepteur et les données vidéo partielles servent de base à la clé utilisée pour chiffrer le premier fragment vidéo. Dans cette implémentation, les autres clés sont obtenues à l'aide de l'adresse MAC du récepteur, de la clé publique et de la clé calculée précédemment. Comme la longueur des paramètres est constante pendant cette procédure, le temps entre chaque clé ne varie pas beaucoup.

Le recours à l'adresse MAC présente par ailleurs l'avantage de limiter la lecture à un appareil spécifique à chaque utilisateur. En fait, la sécurité est ainsi davantage renforcée, car pour lire une vidéo, il faut disposer au préalable des appareils correspondants, en plus des autres mesures de sécurité basées sur les attributs de l'utilisateur que nous avons décrits.

Temps de cryptage des morceaux de vidéo

Dans cette section, nous avons parlé du temps nécessaire pour crypter chaque morceau. En revanche, les modules de traitement vidéo divisent le fichier vidéo en morceaux pour chaque trame complète, l'implémentation assume, dans cet exemple, que la taille des morceaux est de 1 Mo.

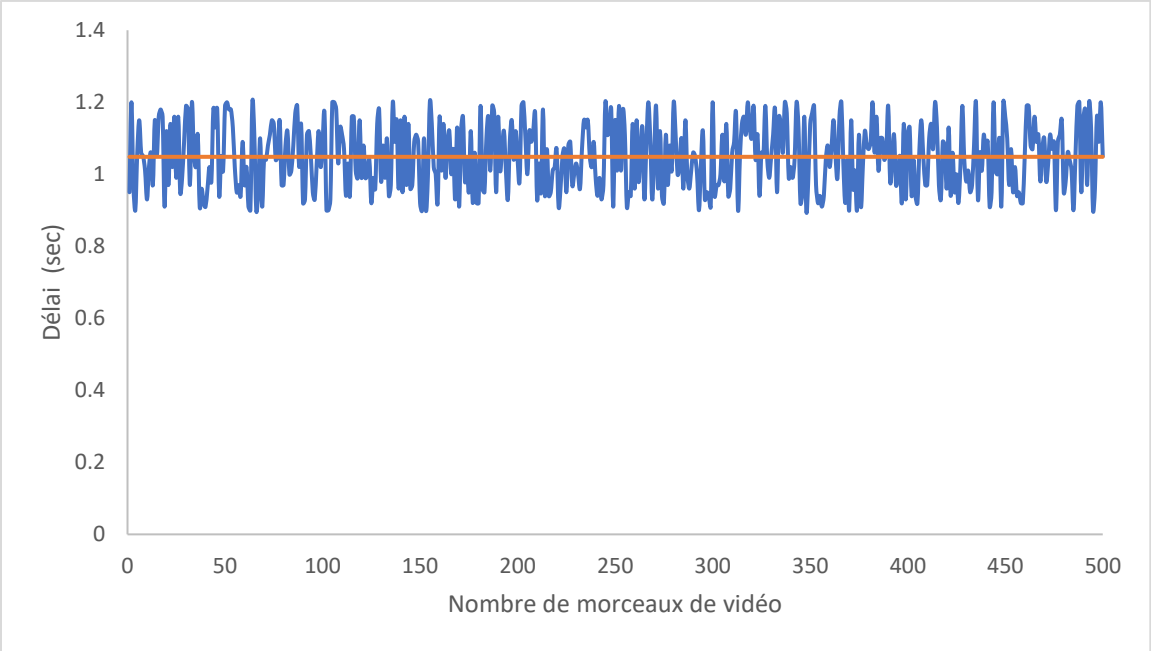


Figure 17 : Temps nécessaire au cryptage des morceaux de vidéo

La figure 17 illustre le délai nécessaire au cryptage de contrôle. Les morceaux de vidéo sont introduits dans le module AES avant d'être transmis. Les morceaux sont pour la plupart identiques et varient entre 1 Mo et 1,2 Mo. Le temps nécessaire au cryptage de chacun d'eux varie également entre 0,9 seconde et 1,2 seconde.

Temps pour crypter les fichiers vidéo

Cette section détaille le temps d'exécution du pipeline, qui commence par la production de morceaux de vidéo et se termine par des morceaux de vidéo chiffrés.

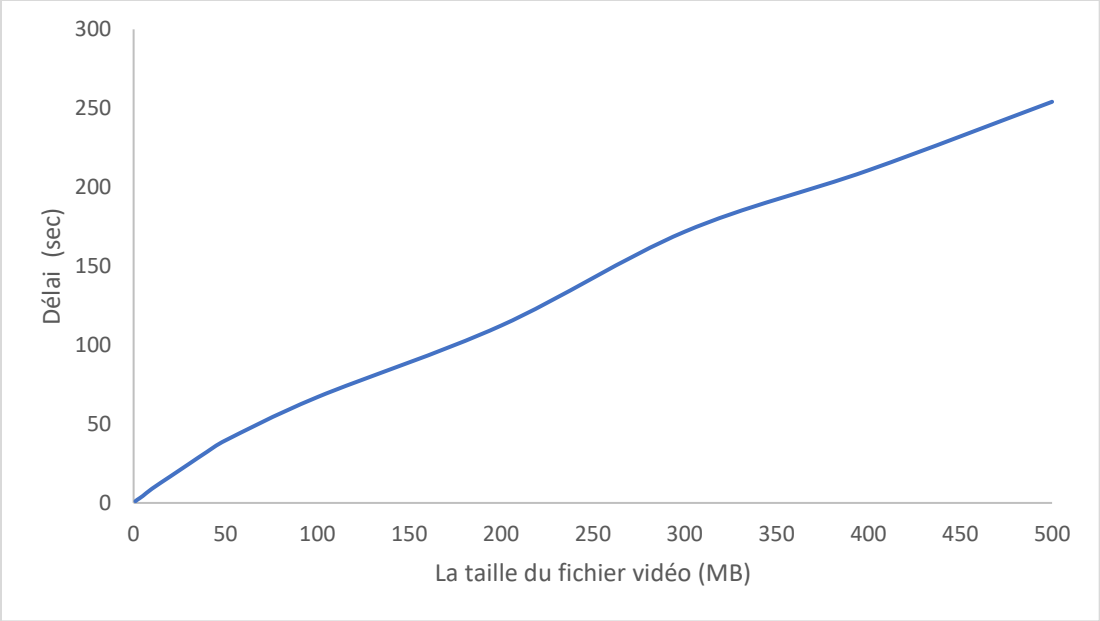


Figure 18 : Délai de bout en bout

Le délai de bout en bout du processus est illustré à la figure 18. Les résultats indiquent que lorsque la taille du fichier augmente, le temps augmente aussi progressivement. Ceci est typique pour toutes les applications. En outre, il est important de noter que la figure 18 illustre le délai ajouté par chaque morceau du clip vidéo. La diffusion et la lecture de la vidéo du côté du récepteur ne sont cependant pas affectées par ce phénomène.

Nombre de clés générées

Le nombre de clés générées dépend de la quantité de morceaux de vidéo générés. Cette statistique a été prise en compte dans l'étude car la solution proposée utilise des technologies à clés multiples pour améliorer la sécurité. Les clés multiples n'ont aucun impact sur l'utilisation de la mémoire car les clés ne sont que momentanément sauvegardées du côté de l'émetteur et du récepteur. De plus, comme chaque clé n'est utilisée qu'une seule fois, une augmentation du nombre de clés n'a pas d'impact sur le délai d'extraction.

Délai de décryptage de bout en bout

Dans cette section, le temps de traitement du pipeline côté récepteur a été examiné. Les morceaux de vidéo chiffrés sont livrés à l'application du récepteur, qui les déchiffre séquentiellement. Les morceaux sont ensuite combinés et affichés sur un appareil. Ici, on a pris en compte le temps que prend ce cycle de traitement. Le support de transmission ayant un impact sur le processus de décryptage, les applications réceptrices doivent attendre d'avoir reçu l'intégralité du morceau avant de le traiter. Si l'on compare le délai au le processus de décryptage prend un peu plus de temps que le temps de cryptage

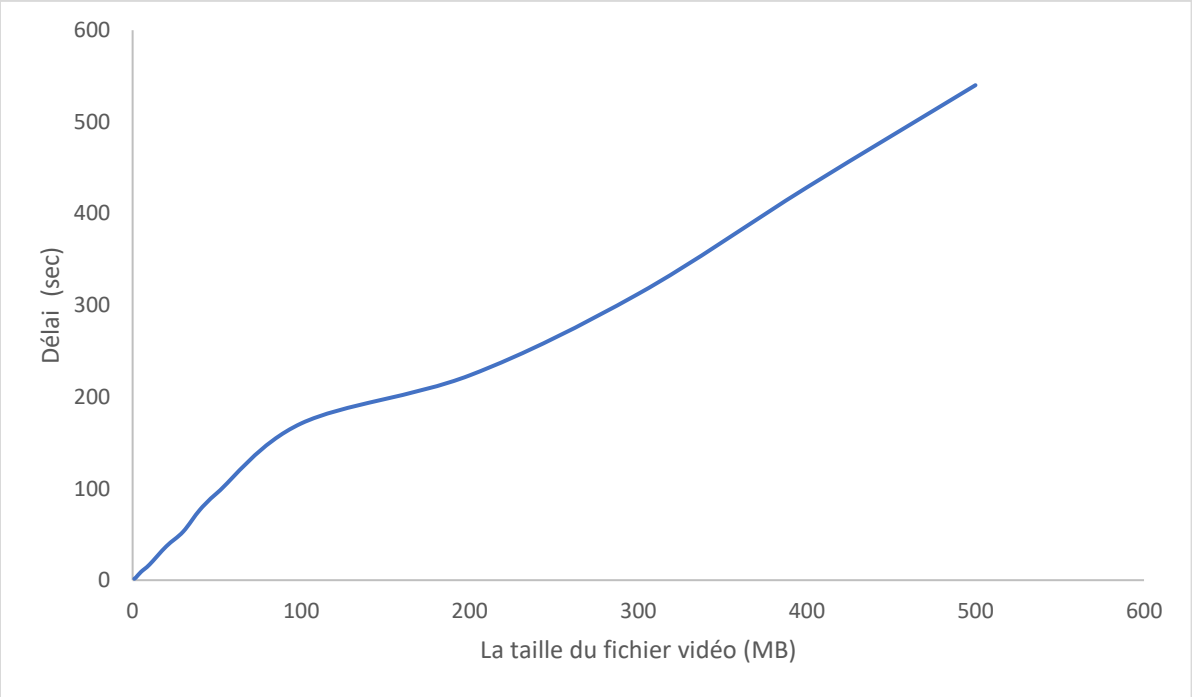


Figure 19 : Délai de décryptage de bout en bout

Conclusion

Cette thèse présente une innovation dans le domaine du cryptage vidéo numérique basé sur l'algorithme de cryptage AES avec un système de génération de clés multiples elliptiques. Un système original de cryptage vidéo utilisant l'algorithme de cryptage AES avec des générateurs elliptiques à clés multiples a été implémenté sur un système Android. Pour améliorer davantage la sécurité du contenu vidéo, nous avons proposé un autre moyen de réduire les risques de perte de la clé qui est généralement envoyée au destinataire par courriel ou par une autre voie de communication. Nous avons donc proposé une nouvelle méthode et un nouvel algorithme basé sur les attributs du récepteur et de la vidéo pour extraire les informations requises du côté du récepteur sans que l'utilisateur final ait à se soucier de la transmission des clés. En conséquence, plusieurs techniques de post-traitement pour surmonter cette faiblesse ont été présentées et adoptées pour cette thèse.

Le système a été décrit en détail en se concentrant sur l'aspect logiciel mis en œuvre sur un système Android. La plateforme de cryptage proposée peut être étendue à n'importe quel format vidéo. Un bref aperçu de la cryptographie a été accompagné dans cette thèse. Enfin, plusieurs tests d'analyse ont été effectués pour prouver que le système proposé répond aux exigences de vitesse et de latence pour le cryptage vidéo en temps réel. De plus, nous avons testé avec succès, sur la plateforme proposée, la lecture vidéo qui utilise le décryptage et la diffusion en temps réel.

La conception de ce système sur une plateforme ASIC pourrait être un travail futur basé sur cette thèse car elle réduit la surface sur puce et fournit un débit plus élevé en utilisant une consommation d'énergie plus faible. En outre, une telle mise en œuvre renforcera la sécurité des

vidéos. L'intégration de cet algorithme de cryptage dans différentes plateformes mérite d'être exploitée.

Chapitre 5 : Conclusion et portée future

Ce chapitre conclut le travail effectué pour cette recherche et décrit son orientation future.

Conclusion

Dans les applications qui impliquent le partage de contenu vidéo, il est extrêmement difficile de sécuriser le contenu vidéo pour respecter les droits d'auteur et éviter la redistribution du contenu. Empêcher la redistribution du contenu vidéo est un défi à l'ère du numérique. C'est pourquoi de nombreux travaux de recherche ont examiné les techniques de cryptographie hybride. Cependant, dans la plupart des cas, le contenu qui est conservé à l'extrémité du récepteur pour permettre l'affichage en temps réel est devenu le goulot d'étranglement.

Le piratage du contenu vidéo a un impact négatif important sur les applications telles que les fournisseurs de médias OTT. La majorité des applications utilisent les méthodes cryptographiques les plus sûres, mais beaucoup d'entre elles ne relient pas les informations relatives au récepteur et à l'appareil pour renforcer la sécurité. Les pirates ont utilisé cette vulnérabilité pour décrypter le contenu de la vidéo.

L'objectif de cette recherche était de fournir une plateforme protégée pour la diffusion de matériel vidéo afin de surmonter les difficultés associées à la communication vidéo, telles que la redistribution du contenu, la préservation des droits d'auteur, la sécurité, la distribution en temps réel et la gestion des clés.

Afin de créer une technique de cryptographie hybride, ce travail prend en compte les algorithmes cryptographiques les plus sûrs, à savoir AES, RSA et ECC. Le principal avantage de l'AES est la variété des longueurs de clé disponibles. La longueur de la clé utilisée pour sécuriser la communication - clés de 128, 192 ou 256 bits - est étroitement liée au temps nécessaire pour

craquer la technique de cryptage. AES est donc nettement plus puissant. Comme le cryptage AES est beaucoup plus rapide, il est parfait pour les logiciels, les microprogrammes et le matériel qui exigent un débit élevé ou une faible latence. RSA permet le cryptage des messages avant leur envoi. En outre, cette technique permet à l'application d'authentifier les notes, en garantissant qu'elles n'ont pas été modifiées ou ajustées pendant leur transit. L'algorithme RSA est actuellement l'une des technologies de cryptage les plus utilisées. Il est extrêmement difficile de craquer la méthode RSA en raison de la complexité des mathématiques utilisées. Le partage des clés publiques avec les utilisateurs est simple en RSA. L'avantage fondamental de la cryptographie à courbe elliptique est que des clés plus courtes sont nécessaires pour garantir un niveau de sécurité spécifique que dans le cas d'un "cryptage normal". Des clés plus courtes peuvent entraîner des économies importantes dans les implémentations matérielles. Le deuxième avantage de la cryptographie par courbes elliptiques est que de nombreuses attaques développées pour la factorisation et la cryptographie par logarithme discret ne fonctionnent pas pour la cryptographie par courbes elliptiques.

Une plateforme innovante et sécurisée a été présentée dans ce travail. La plateforme utilise la méthode cryptographique proposée pour crypter et décrypter le fichier vidéo. Une méthode de cryptographie hybride à clés multiples a été développée dans ce travail de recherche. Elle sécurise le contenu vidéo en utilisant RSA, l'équation ECC et AES. Voici quelques-unes des qualités distinctives de la plateforme proposée :

- Le destinataire n'a pas accès aux clés.
- L'adresse MAC de l'appareil est utilisée pour empêcher le partage des ressources.
- Les fichiers vidéo décryptés sont lus sur un écran sans être stockés du côté du récepteur.
- La vidéo est décomposée en morceaux et chiffrée avec différentes clés.

- Pour sauvegarder le contenu vidéo variable d'un fichier vidéo à l'autre, chaque récepteur disposera d'une clé unique.

La solution à clés multiples qui a été présentée sépare la vidéo en de nombreux petits morceaux, puis crypté chaque morceau avec une clé différente. Les récepteurs n'ont pas accès à ces clés. En utilisant les morceaux cryptés qu'elle a reçus, l'application du récepteur génère la clé correspondante à chaque morceau crypté. Le processus de décryptage est lancé par l'application du récepteur dès que les morceaux de vidéo sont reçus, car la méthode proposée est dynamique et automatique.

L'application a été créée du côté du récepteur pour la plateforme Android. Un programme côté serveur basé sur java a été créé pour mettre en œuvre la méthode suggérée et les tests. Sur la base des tâches prises en compte dans la stratégie suggérée, des modules distincts ont été développés. Ensuite, des fichiers vidéo de différentes tailles ont été utilisés pour tester l'application. Les résultats ont montré que le délai de l'application est raisonnable et prend en charge la communication vidéo en temps réel. L'approche hybride utilise les algorithmes bien établis AES, RSA et ECC. La sécurité du contenu vidéo est améliorée par ces approches cryptographiques.

La plateforme proposée présente un avantage pour les applications de vidéo à la demande, car elle utilise les informations d'identification et les informations sur le dispositif du destinataire pour crypter et diffuser sur le réseau tout en protégeant le contenu vidéo avec une clé dynamique. Toutes les clés dérivées et le contenu crypté sont également configurés dans le programme pour être temporairement sauvegardés sur le dispositif du destinataire. Grâce à cette technique innovante proposée, les droits d'auteur peuvent être efficacement protégés en bloquant le transfert

des contenus. Ce soutien permet d'augmenter le nombre d'abonnés au contenu vidéo et, en fin de compte, les revenus.

Portée future

La méthode actuelle s'attaque uniquement à la vidéo à la demande, qui ne représente qu'une petite partie du spectre de l'information. L'information est diverse et différents types de données, comme l'audio, la vidéo, les graphiques, etc. sont utilisés dans les transactions d'aujourd'hui.

Le service de cryptage traitera tous ces formats de données. Afin d'accommoder les communications sécurisées complètes d'aujourd'hui, ces caractéristiques doivent être prises en compte dans le processus de transformation cryptographique.

Dans la continuité de l'étude, les actions suivantes pourraient être menées, ce qui pourrait conduire au développement d'un algorithme meilleur et plus flexible, permettant une sécurité accrue et une diversité d'applications.

1. Pour identifier les forces et les failles de l'algorithme, celui-ci peut être soumis à une cryptanalyse rigoureuse à l'aide d'outils spécialisés dans ce but. Cela permettra d'identifier les endroits où des modifications pourraient être apportées.
2. Le système peut être modifié pour traiter du texte riche, de la vidéo et de la parole. Cela fournira au système une fonctionnalité complète de cryptage multimédia, permettant le cryptage de pages Web modernes.

Annexe: Papier Journal-IEEE ACCESS

“Une nouvelle technique hybride de cryptographie à clés multiples pour la communication vidéo”

“A Novel Hybrid Multikey Cryptography Technique for Video Communication”

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

A Novel Hybrid Multikey Cryptography Technique for Video Communication

YOUCEF FOUZAR[1, 2], AHMED LAKHSSASSI[1, 3], and RAMAKRISHNA M[4]

[1]University in Gatineau, Quebec, Canada
[2](e-mail: fouy03@uqo.ca)
[3](e-mail: ahmed.lakhssassi@uqo.ca)
[4]Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal, India (e-mail: ramakrishna.m@manipal)

Corresponding author: Youcef Fouzar (e-mail: fouy03@uqo.ca).

ABSTRACT Online Video Streaming is becoming common in daily routine for entertainment. This thesis reports an original work on the software realization of video encryption and decryption using continuous systems based on the Elliptic Curve Cryptography technique as pseudo-random encryption key generators. This method generates multi keys to encrypt and decrypt smaller chunks of the video file. The security of the proposed system is analyzed exhaustively, and the performance is compared with other reported systems, showing superiority in terms of performance and security. The thesis focuses more on the hardware and circuit aspects of the system design. The system is realized on Xilinx Vetrix-4 FPGA with hardware parameters and throughput performance outperforming conventional encryption systems. The same system is also realized on a smart device with an Android platform.

INDEX TERMS Asymmetric Key Cryptography, Multikey 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 neededto handle the issues of video communication. Here, themost 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 cryptography is the one most used in the 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 implementing

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 securityof the Internet. Hence, many research works 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. This paper has developed a novel method that uses asymmetric key cryptography to encrypt video chunks. This work's contribution involves designing and developing a novel equation-based multikey encryption technique and video attribute-based decryption key generation method.

This works aims at developing a multikey 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 for generating encrypted video chunks.
- Multikey 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

U. Zia *et al.* [20] proposed a pseudo-random 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. H. Kezia *et al.* [21] developed a novel video encryption scheme based on chaotic maps. Here, first, the video sequence to be encrypted 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 multikey concept is used to improve the security of the cryptosystem against attacks.

M. A. 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. R. Imam *etal.* [23] reviewed RSA-based cryptographic techniques and suggested the suitability of the crypto techniques for various applications.

N. 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. Z. Chen *et al.* [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.

R. Hegde *et al.* [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.

S. H. Murad *et al.* [29] have studied two-tier and three-tier hybrid cryptography model applied for cloud security. s. K. Ghosh *et al.* [30] proposed a hybrid method to achieve confidentiality and security of the data in Internet. J. Zhang *et al.* [31] proposed a method for the video data that is encoded using layered coding method. L. Yu *et al.* [32] have discussed the applications of the hybrid encryption algorithm in software securities. The work discusses the uses of hybrid method in enhancing the security basically in video surveillance software.

M. A. Khan *et al.* [33], in order to accomplish the Hybrid encryption technique, data encryption techniques using 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 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 is the basis of this work.

C. L. 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 EncryptionStandard (AES) and ElGamal with Double Playfair Cipher (DPC). J. Dave *et al.* [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.

M. Hamdi et al. [36] have proposed a hybrid encryption algorithm based on block and stream ciphers using chaotic systems. The proposed scheme adopts two main operations one to generate 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 block cipher.

D. Das [38] explored a unique hybrid model byleveraging power of automation for security tests at the Video Acquisition/Aggregation level and amalgamating the best practices from traditional security tests done at user Video Application level. This hybrid methodology coversall video streaming value chain phases, from origin to playback. Therefore, it achieves maximum test coverage across multiple playback devices under multiple workload conditions. In this methodology, we deploy real-world conditions, including latency, delay, and concurrency.

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 multikey cryptography technique to improve security in video communication. The proposed method uses RSA and ECC as basics for achieving 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 multikey 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 multikey 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 advantages of the hybrid crypto techniques, the proposed uses RSA, ECC and AES techniques
- improving the security by multikey encryption technique
- increasing the security with the video chunks
- reduce multikey management using automatic key generation methods

The goals are achieved using the proposed key generations model shown in Figure 1.

The process is initiated by passing video data to the key generation module. The module requires the receiver's Public Key $R_{P\ key}$ and MAC address R_{mac}; these improve the uniqueness of the keys that are generated in the process. These attributes are required throughout the video communication; hence it is stored on the sender 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 2 discusses the steps involved in the key generation. The encryption of video chucks starts with the step of 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 $R_{P\ key}$
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} is used for generating the subsequent key. Here, the method uses Public Key $R_{P\ key}$ 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.,

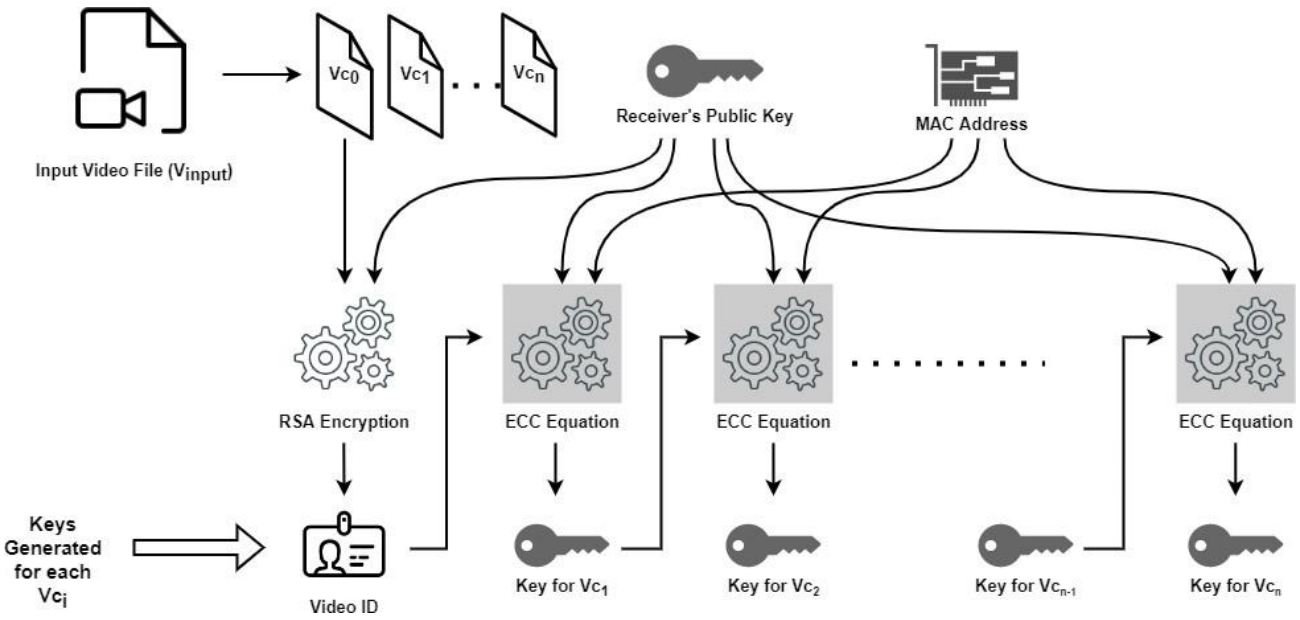


FIGURE 1: Block Diagram of Proposed Key Generation Technique

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

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 for the first video chuck only. 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 multikey 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 as well as video streaming.

For the purpose of streaming, the sender devices store the video content and video metadata. When a receiver requests to stream a video file, the sender application starts up. Before sending the video content to the recipient, the sender, an authorised video distributor, verifies the recipient's identity. A database is used to maintain receiver information. Information about the recipient gathered during the sign-up process. where 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 process employs the sender's public key and the receiver's private key. 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.

An implementation of a database is made necessary by 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 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

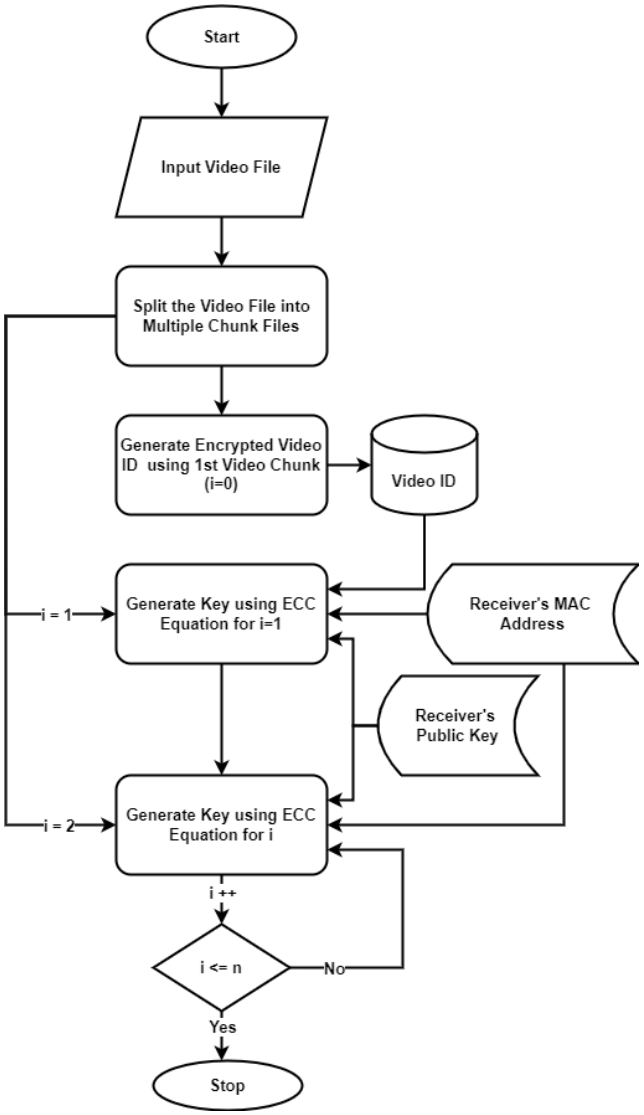


FIGURE 2: Block Diagram of Proposed Key Generation Technique

account the receiver's public key and MAC address.

The video file is read by the sender side module, which also creates the video chunks. The video chunks in this instance were generated 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 length of 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. The processing time involved in the video processing module is therefore demonstrated by this investigation. The chunk size is read by the FFMpeg utility from the sender. The chunk is then divided into the nearest full frame, resulting in a chunk size that is determined by user input.

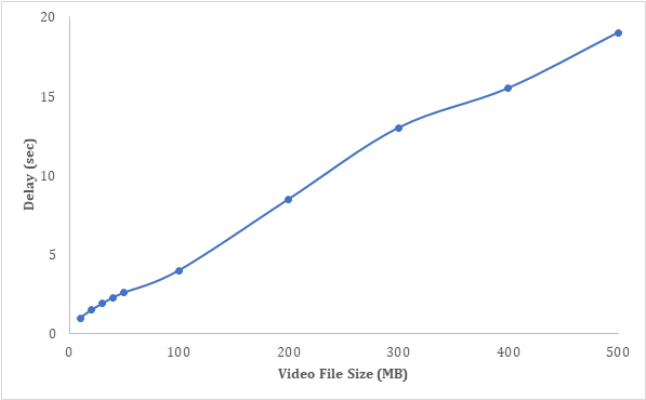


FIGURE 3: Time taken by the video processing module to split the video into chunks

The length of time needed by the video processing module to split the video file into several chunks is depicted in Figure 3. The outcome shows that the latency increasesgradually 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 the high security, however, this is necessary.

B. TIME TO GENERATE THE KEYS

To determine the impact of the multikey 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.

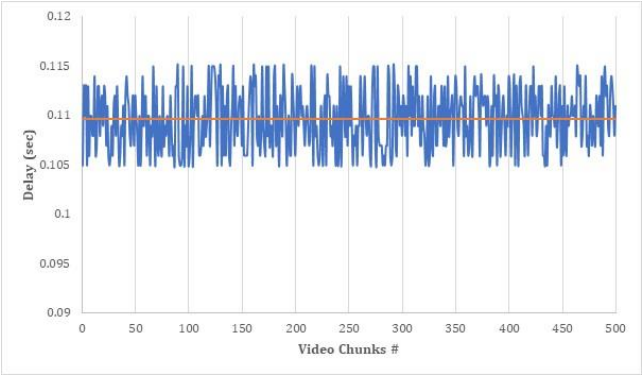


FIGURE 4: Delay in generating multiple keys

Figure 4 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.

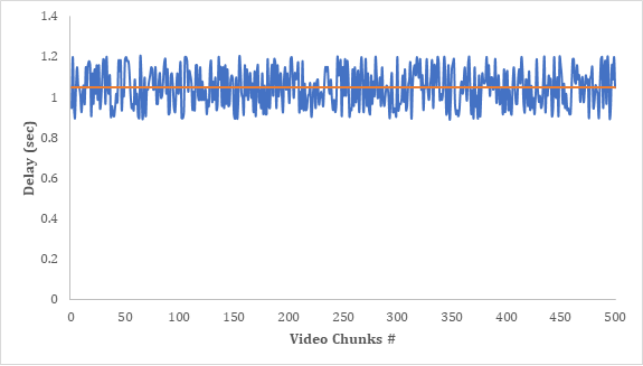


FIGURE 5: Time taken to encrypt the video chunks

Figure 5 depicts the delay involved in the check encryption. The video chunks are fed into the AES module before being transmitted. The chunks are mostly of same that it is vary between 1MB and 1.2MB, the time taken to encrypt the each is also vary 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.

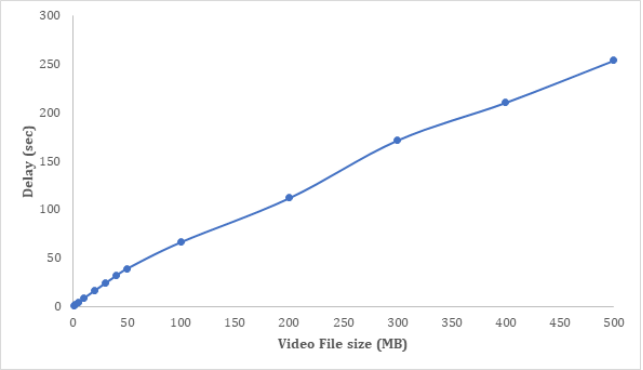


FIGURE 6: End-to-end encryption delay

The process's end-to-end latency is depicted in Figure 6. The findings indicate that as the file size increases, the time also does increase progressively. This is typical for all applications. Furthermore, it is significant to note that Figure 6 depicts the added delay by each chunk of the video clip. The video streaming and playback at the receiver side are unaffected by this, though.

E. NUMBER OF KEYS GENERATED

The number of keys generated depends on the quantity of generated video chunks. This statistic has been taken into consideration for the study because the suggested solution uses multiple key technologies to provide improved security. Multiple keys 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 utilised 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 taken into account how much time this processing cycle takes. Because the transmission medium has an impact on the decrypting process, receiver apps must wait until they have received the entire chunk before processing it. Comparing the delay to encryption, this might make it longer.

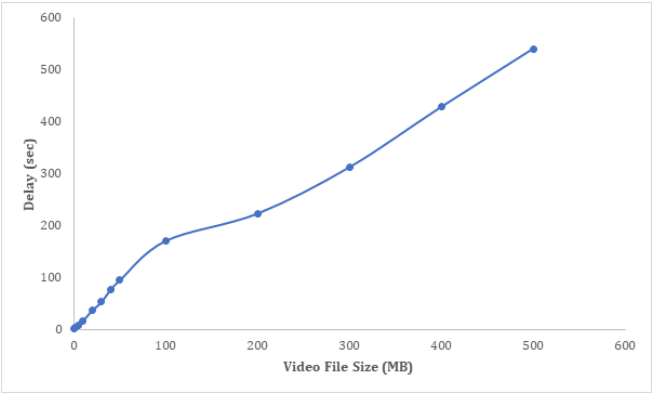


FIGURE 7: End-to-end Decryption delay

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 multikey cryptography method has developed in this work. It secures the video contents using RSA, the ECC equation, and AES.

The multikey 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. Using the encrypted chunks, it has received,the receiver application generates the key. The decryption process is started by the receiver application as soon as the

video chunks are received because the suggested method is dynamic and automatic.

The application was created at the receiver side for the Android platform. A java-based server-side program hasbeen created to implement the suggested method and testing.On the basis of 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 makes use of the well-established AES, RSA, and ECC algorithms. The security of the video content is improved by these cryptography approaches.

The suggested platform has an advantage for video on-demand applications since it uses the recipient's credentials and device information to encrypt and stream across the network while safeguarding the video contents with a dynamic key. All derived keys and encrypted contentare also configured in the program to be temporarily savedat the recipient device. With this help, copyright can be effectively safeguarded by blocking the transfer of the contents. This support growing the number of subscribers tovideo content and, ultimately, revenue.

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 Transactions on Broadcasting*, vol. 64, no. 1, pp. 11–25, 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, January 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," 4 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," 2021. [Online]. Available: https://dx.doi.org/10.21227/s7x7-wd58

[5] "Cisco Annual Internet Report - Cisco Annual Internet Report (2018–2023) White Paper." [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." [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," *Proceedings of the 7th Conference on Emerging Networking EXperiments and Technologies, CoNEXT'11*, 2011.

[8] X. Huang, D. Arnold, T. Fang, and J. Saniie, "A chaotic-based encryption/decryption system for secure video transmission," in *2021 IEEE International Conference on Electro Information Technology (EIT)*, 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, no. 1, dec 2008.

[10] A. Murtaza, S. J. Hussain Pirzada, and L. Jianwei, "A new symmetric key encryption algorithm with higher performance," in *2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, 2019, pp. 1–7.

[11] S. Kansal and M. Mittal, "Performance evaluation of various symmetric encryption algorithms," in *2014 International Conference on Parallel, Distributed and Grid Computing*, 2014, pp. 105–109.

[12] S. Kumar, M. S. Gaur, P. Sagar Sharma, and D. Munjal, "A novel approach of symmetric key cryptography," in *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, 2021, pp. 593–598.

[13] M. Dworkin, E. Barker, J. Nechvatal, J. Foti, L. Bassham, E. Roback, and J. Dray, "Advanced encryption standard (aes)," 2001-11-26 2001.

[14] Y. Shen, Z. Sun, and T. Zhou, "Survey on asymmetric cryptography algorithms," in *2021 International Conference on Electronic Information Engineering and Computer Science (EIECS)*, 2021, pp. 464–469.

[15] S. Kumar, B. K. Singh, Akshita, S. Pundir, S. Batra, and R. Joshi, "A survey on symmetric and asymmetric key based image encryption," in *2nd International Conference on Data, Engineering and Applications (IDEA)*, 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, p. 120–126, feb 1978. [Online]. Available: https://doi.org/10.1145/359340.359342

[17] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, Jan. 1987.

[18] A. J. Menezes and S. A. Vanstone, "Elliptic curve cryptosystems and their implementation," *Journal of Cryptology 1993 6:4*, vol. 6, pp. 209–224, 9 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 and Cryptography 2000 19:2*, vol. 19, pp. 173–193, 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,"*SN Applied Sciences*, vol. 4, pp. 1–17, 2 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 *2008 16th International Conference on Advanced Computing and Communications*, 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. 52 018–52 027, 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. 155 949–155 976, 2021.

[24] N. Sen, R. Dantu, J. Vempati, and M. Thompson, "Performance analysis of elliptic curves for real-time video encryption," in *2018 National Cyber Summit (NCS)*, 2018, pp. 64–71.

[25] S. C. Iyer, R. Sedamkar, and S. Gupta, "A novel ideaon multimedia encryption using hybrid crypto approach," *Procedia Computer Science*, vol. 79, pp. 293–298, 2016,proceedings of International Conference on Communication, Computing and Virtualization (ICCCV) 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 *2012 International Conference on Computing, Communication and Applications*, 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, p. 169676, 2022. [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," *Computer Standards & Interfaces*, vol. 48, pp. 173–182, 11 2016.

[29] S. H. Murad and K. H. Rahouma, "Implementation and performance analysis of hybrid cryptographic schemes applied in cloud computing environment," *Procedia Computer Science*, vol. 194, pp. 165–172, 2021, 18th International Learning & Technology Conference 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 *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, 2020, pp. 1–5.

[31] J. Zhang and X. Gao, "A hybrid encryption scheme for scalable video coding based on h.264," in *2013 5th International Conference on Intelligent Networking and Collaborative Systems*, 2013, pp. 708–711.
[32] L. Yu, Z. Wang, and W. Wang, "The application of hybrid encryption algorithm in software security," in *2012 Fourth International Conference on Computational Intelligence and Communication Networks*, 2012, pp. 762–765.
[33] M. A. Khan, K. K. Mishra, N. Santhi, and J. Jayakumari, "A new hybrid technique for data encryption," in *2015 Global Conference on Communication Technologies (GCCT)*, 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, 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 Nature Singapore, 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, p. 1847–1858, feb 2021. [Online]. Available: https://doi.org/10.1007/s00500-020-05258-z
[37] S. Chen, S. Yu, J. Lü, G. Chen, and J. He, "Design and fpga-based realization of a chaotic secure video communication system," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2359–2371, 2018.
[38] P. Yu, N. Zhang, S. Zhang, and Q. Wang, "Security mechanism of video content integrated broadcast control platform under triple play," in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 2017, pp. 1–5.
[39] D. Das, "Automated security testing framework for validating content rights on video streaming devices," in *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, 2019, pp. 516–521.
[40] J. K. Joshi, D. S. Bais, and A. N. Dubey, "An efficient and secure method for quality video streaming in mobile ad-hoc network," in *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, 2015, pp. 75–79.
[41] J. K. Liu, M. H. Au, W. Susilo, K. Liang, R. Lu, and B. Srinivasan, "Secure sharing and searching for real-time video data in mobile cloud," *IEEE Network*, vol. 29, no. 2, pp. 46–50, 2015.
[42] E. Lin, A. Eskicioglu, R. Lagendijk, and E. Delp, "Advances in digital video content protection," *Proceedings of the IEEE*, vol. 93, no. 1, pp. 171–183, 2005.
[43] R. Ahuja, A. Kaur, V. Lamba, V. Kukreja, A. Agarwal, and M. Sharma, "Securing copyright of digital video by exploiting quantized dc coefficients," in *2021 6th International Conference on Signal Processing, Computing and Control (ISPCC)*, 2021, pp. 423–427.
[44] L. Mou, "Ownership identification and signaling of multimedia content components," in *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, 2018, pp. 212–213.
[45] J. Ning, J. Chen, K. Liang, J. K. Liu, C. Su, and Q. Wu, "Efficient encrypted data search with expressive queries and flexible update," *IEEE Transactions on Services Computing*, vol. 15, no. 3, pp. 1619–1633, 2022.