N° d'ordre :

REPUBLIQUE ALGERIENNE DEMOCRATIQUE & POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR & DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE DJILLALI LIABES
FACULTE DES SCIENCES EXACTES
SIDI BEL ABBÈS

THESE DE DOCTORAT

Présentée par

OUAMRI Mokhtar

Spécialité : *Informatique*
Option : *Réseaux des systèmes informatiques*

Intitulée

Sécurité et compression de l'information multimédia

Soutenue le 07/01/2016
Devant le jury composé de :

Président :	BOUKLI HACENE Sofiane	MCA, UDL - Sidi Bel Abbés.
Examinateurs :	BELALEM Ghalem	Professeur, l'université d'Oran 1.
	ELBERRICHI Zakaria	Professeur, UDL - Sidi Bel Abbés.
	BENMAMMAR Badr	MCA, l'université de Tlemcen.
	BOUCHIHA Djelloul	MCA, centre Universitaire de Naama.
Directeur de thèse :	FARAOUN Kamel Mohammed	Professeur, UDL - Sidi Bel Abbés.

Année universitaire *2015/2016*

Dédicaces

A mon cher père Hocine
Et ma chère mère Fatima
Pour l'éducation et le grand amour dont ils m'ont accordé durant toutes ma vie.
A mes grandes mère Bakhta et Fatna
A mes grands père Amar et Mohammed
A mes sœurs assia et latifa
A l'âme de mon oncle Mokhtar ouamri (martyre de la guerre nationale)
A tous ceux que j'aime.
Je dédie ce travail

Mokhtar OUAMRI

Remerciements

Je remercie, au premier lieu, mon Dieu qui m'a offert et préservé une bonne santé et qui m'a entouré de sa bienveillance et sa grâce. Je le remercie également de m'avoir confié à des gens respectueux, responsables et scientifiques pendant ces années de thèse.

Je ne pense pas que quelques mots de remerciements puissent suffire pour exprimer le sentiment de profonde gratitude et de reconnaissances que j'éprouve à mon directeur de thèse Monsieur Pr. FARAOUN Kamel Mohammed, Professeur à UDL Sidi Bel-Abbés (Algérie), pour m'avoir encadré avec vigilance, disponibilité totale et une clairvoyance remarquable pour ces travaux. Qu'ils trouvent ici mes remerciements les plus sincères.

Je remercie très sincèrement, les membres de jury d'avoir bien vouloir accepter de faire partie de la commission d'examinateur.

Je remercie Monsieur Dr. BOUKLI HACENE Sofiane, Maitres de conférences classe A à UDL Sidi Bel-Abbés (Algérie), pour l'honneur qu'il me fait d'avoir accepté d'être le président du jury.

Je remercie également Monsieur Pr. BELALEM Ghalem, Professeur à l'université d'Oran 1 (Algérie), Monsieur Dr. BENMAMMAR Badr, Maitre de conférences classe A à l'université de Tlemcen (Algérie), Monsieur Dr BOUCHIHA Djelloul Maitre de conférences classe A à centre universitaire de Nàama,et Monsieur Pr. ELBERRICHI Zakaria, Professeur à UDL Sidi Bel-Abbés (Algérie) d'avoir accepté d'être les examinateurs de ma thèse. Je les remercie pour l'attention avec laquelle ils ont lu et évalué ce travail.

J'adresse mes remerciements à mes parents et mes sœurs pour leur soutien durant toute ma vie.

J'adresse également nos remerciements, à tous nos enseignants, qui nous ont données les bases de la science.

Enfin, je remercie tous ceux qui est de prés ou de loin ont contribués par leur encouragement et leur aide a la réalisation de ce travail.

Introduction

L'information multimédia occupe une place incontournable dans notre vie quotidienne avec un succès florissant et incontestable dans les marchés actuels de la technologie numérique. Elle qualifie en fait, l'intégration de plusieurs moyens de représentation de l'information telle que le texte, l'image, la vidéo, et l'interactivité. De même, elle occupe une place dominante et omniprésente dans le marché de la technologie numérique, où elle est déployée dans une large gamme d'applications allant de jeu vidéo à la télévision numérique, de bureautiques aux applications réseaux et mobiles, et aussi dans des applications relatives à la communication numérique et à l'intelligence artificielle. La révolution multimédia est toujours en pleine expansion, et elle connait en conséquence des nouveautés et d'évolution en normes émergentes de codage pour répondre aux larges besoins expansifs de consommateurs et/ou constructeurs.

La vidéo est l'une des informations multimédias les plus utilisées. On la trouve dans des applications très variés en ou hors ligne comme la vidéo conférence, VOD (vidéo à la demande), et dans les réseaux sociaux comme celles de streaming. Une communication sûre déployant la vidéo numérique nécessite le passage par des étapes pionnières comme la compression (ou le codage de source) pour réduire la quantité transmise et d'extraire l'information pertinente à transporter, le codage pour combattre les erreurs lors de transmission et la protection de l'information par des mesures de sécurité logicielle et/ou matérielle comme la signature numérique, la stéganographie, le tatouage numérique, et la cryptographie par des approche de chiffrement.

Le chiffrement (le cryptage) est l'un des mesures de sécurité logicielle qui est employé massivement afin de garantir des qualités primordiales comme la confidentialité, l'intégrité et l'authentification. Il permet de transformer le contenu informatif d'une vidéo claire en inintelligible au moyen d'un algorithme (publique) et d'une clé secrète. Sans ce dernier moyen, l'utilisateur ne pourra jamais accéder à l'information originale de la vidéo car le contenu de la vidéo chiffrée sera en conséquence illisible. Le chiffrement peut s'appliquer pour dégrader la qualité de la vidéo, changer entièrement le contenu, ou modifier carrément le format du flux.

La conception d'un cryptosystème pour le chiffrement de vidéo repose sur l'étude de processus de compression utilisé et le format de codage adopté. Le défi à surmonter dans cette thèse est de proposer un schéma robuste pour le chiffrement sélectif de vidéo durant son codage entropique. Nous avons choisi la norme de codage video HEVC (High Efficiency Video Coding) qui est une norme récente standardisée en Avril 2013, et prévue a être le futur codec de l'ère Ultra HD.

Le contenu de ce mémoire est organisé comme suit :

Dans le premier chapitre, on découvrira un petit historique sur l'évolution de l'information. Après, on abordera la théorie de l'information pour se rapprocher au codage. Et on terminera avec la compression au sens général.

Puisque la vidéo n'est qu'un défilement d'une séquence d'images animées, nous commencerons par donner dans le chapitre II, un aperçu sur la compression d'images numériques. Après, nous passerons à la compression de vidéo en expliquant les différentes étapes de compression liées allant de la prédiction au codage entropique. Aussi, on va citer quelques normes de compression vidéo normalisées depuis les deux groupes de normalisation ISO et ITU-T.

Dans le chapitre III, nous aborderons les exigences adressés par la communauté scientifique pour le lancement du projet HEVC, un bref historique sur l'évolution de HEVC jusqu'à son standardisation, et les différentes étapes de codage associés, et les applications et les extensions liées.

Dans le chapitre IV, nous commencerons par donner des définitions et vocabulaires relatives au domaine de la cryptographie moderne. Tout en restant général, on abordera également les algorithmes de chiffrement symétrique et asymétrique en citant quelques algorithmes classiques et populaires. Après, on passera aux techniques de chiffrement appliquées pour la protection de la vidéo numériques. Ces techniques dépendent inévitablement de la norme de codage de vidéo, et elles peuvent être appliqué avant, durant, ou après la compression. Finalement, on terminera par une conclusion pour illustrer les défis à surmonter pour protéger la norme récemment standardisée HEVC.

L'objectif de chapitre V est de présenter une nouvelle approche de chiffrement sélectif conforme à la norme HEVC. Elle permet en outre de générer un flux binaire décodable selon la dernière version de document standardisé de HEVC. Notre approche consiste à choisir protéger les signes et les codes de type Golomb-Rice de QTCs non nuls, et les chiffrer a l'aide de cryptosystème AES en mode opératoire CBC. Après avoir présenté le codage entropiques des QTCs selon les dernières modifications achevées en HEVC, nous allons exposer la problématique de chiffrement par mentionner les inconvénients et les lacunes observées dans les travaux antérieurs. Par la suite, nous allons expliquer notre approche en détail. La validation de notre approche est évaluée par l'illustration de plusieurs résultats expérimentaux.

Dans le dernier chapitre, nous allons proposer une autre nouvelle approche de chiffrement sélectif s'appliquant aussi durant le module de codage entropique de HEVC, où nous allons protéger les données fréquentielles (amplitudes et signes de QTCs) selon le dernier codage entropique publié de QTCs. Premièrement, nous allons commencer par donner une brève description de ce codage. Après, nous allons exposer la problématique par mentionner les différences qui existent entre les codages antérieurs et présents de QTCs. Ensuite, nous allons expliquer notre approche de chiffrement sélectif. Notre approche est soumise à une évaluation par des tests expérimentaux.

Résumé :

L'information multimédia occupe une place incontournable dans notre vie quotidienne avec un succès florissant et incontestable dans les marchés actuels de la technologie numérique. La vidéo est l'une des informations multimédias les plus utilisées. On la trouve dans des applications très variés en ou hors ligne comme la vidéo conférence, VOD (vidéo à la demande), et dans les réseaux sociaux comme celles de streaming. Une communication sûre déployant la vidéo numérique nécessite le passage par des étapes pionnières comme la compression (ou le codage de source) pour réduire la quantité transmise et d'extraire l'information pertinente à transporter, le codage pour combattre les erreurs de transmission et la protection de l'information par des mesures de sécurité logicielle et/ou matérielle comme la signature numérique, la stéganographie, le tatouage numérique, et la cryptographie par des approche de chiffrement.

HEVC (High Efficiency Video Coding) est la dernière norme de codage vidéo. Elle apporte une architecture hybride de codage permettant ainsi la compression de vidéos à haute définition. En effet, cette norme est attendue à être le futur codec de l'ère Ultra HD. Comme la vidéo HEVC représente le résultat d'une compression de grande quantité de données visuelles, le maintien de la taille de flux binaire compressé s'avère inévitable lors de la conception d'un système de crypto-compression pour la protection de vidéos HEVC. Et ceci pourra être atteint par l'inclusion de module de chiffrement durant CABAC (Context-adaptive binary arithmetic coding) qui est le seul codeur entropique utilisé depuis HEVC.

Dans cette thèse, nous avons proposé deux approches de chiffrement sélectif pour les vidéos HEVC, où nous avons chiffré des éléments syntaxiques relatives aux coefficients fréquentiels quantifiés. Dans notre première approche, les codes de Golomb-Rice récemment introduits dans HEVC et les signes de coefficients non-nuls sont protégés. Alors dans la deuxième approche, nous avons choisi tous les codes utilisés pour le codage de coefficients non-nuls avec ses signes pour le chiffrement dans un contexte bien étudié. Les deux approches proposées génèrent des flux binaires HEVC cryptés conformes à la norme de codage avec une taille identique aux flux binaires clairs. En conséquence, les contributions présentées dans cette thèse constituent en effet un premier pas vers la sécurité de la norme HEVC.

Mots clés : *HEVC, chiffrement sélectif, CABAC, codage vidéo.*

Abstract :

The multimedia information occupies an essential place in our daily life with a booming success in current digital technology markets. The video is one of the most used multimedia information, and it is found in many different applications on/offline such as video conference, VOD (video on demand) and in social networks such as streaming. Secure communication deploying digital video requires passing by pioneer steps such as compression (or source coding) to reduce the amount of transmitted information, coding to combat transmission errors, and protection of communicated information by security measures such as digital signature, steganography, watermarking and cryptography by encryption approach.

HEVC (High Efficiency Video Coding) is the latest video coding standard. It provides a hybrid coding architecture allowing the compression of high definition videos. Indeed, this standard is expected to be the future codec of Ultra HD era. As the video HEVC represents the result of a compression of large amount of visual data, maintaining the compressed bitstream size is unavoidable in the design of a crypto-compression system for the protection of HEVC videos. And this can be achieved by the inclusion of encryption module during CABAC (Context-adaptive binary arithmetic coding) which is the only entropy coder used by HEVC.

In this PhD thesis, we proposed two selective encryption approaches for HEVC videos, where we encrypted syntax elements related to quantized frequency coefficients. In our first approach, Golomb-Rice codes recently introduced into HEVC and signs of non-zero coefficients are protected. While in the second approach, we selected all codes used for coding non-zero coefficients with signs for encryption in a context well-defined. The two proposed approaches generate a compliant encrypted HEVC bitstream with a same size as the original bitstream. Consequently, the contributions presented in this thesis are indeed a first step towards the security of the HEVC standard.

Keywords: HEVC selective encryption, CABAC, video coding.

ملخص :

تحتل معلومات الوسائط المتعددة مكانا أساسيا في حياتنا اليومية و نجاحا مزدهرا في أسواق التكنولوجيا الرقمية الحالية .الفيديو هو واحد من معلومات الوسائط المتعددة الأكثر استخداما، بدليل وجوده في شتى التطبيقات الرقمية مثل نظام مؤتمرات الفيديو، الفيديو حسب الطلب والشبكات الاجتماعية .الإتصال الآمن بالفيديو الرقمي يتطلب اجتياز خطوات رائدة مثل الضغط أو ترميز المصدر الذي يهدف إلى تقليل كمية المعلومات المرسلة، الترميز لمكافحة أخطاء الإرسال، وحماية المعلومات المرسلة بحلول أمنية مثل التوقيع الرقمي، إخفاء المعلومات، الوشم، والتشفير.

HEVC (ترميز الفيديو عالي الكفاءة) هو أحدث معيار مستحدث لترميز الفيديو. ويوفر بنية ترميز هجينة تسمح بضغط أشرطة الفيديو HD. ومن المتوقع أن يصبح هذا المعيار أيقونة في عصر فيديوهات عالية الوضوح. بما أن هذا النوع المميز من الفيديو يمثل نتيجة ضغط كمية كبيرة من البيانات البصرية، فإن الحفاظ على حجم الملف المضغوط أمر لا مفر منه في تصميم نظام تشفير-ضغط لحماية ملفات الفيديو. وهذا لا يمكن تحقيقه إلا عن طريق إدراج وحدة التشفير خلال وحدة الكاباك (الترميز الحسابي الثنائي المكيف حسب السياق) الذي يعد النظام الترميز الأنثروبي الوحيد المستخدم من قبل HEVC.

في هاته الأطروحة ، اقترحنا مقاربتين للتشفير الانتقائي للفيديو HEVC، حيث أننا قمنا بتشفير عناصر بنيوية متعلقة بترميز معاملات ترددية. في المقاربة الأولى، قمنا بحماية رموز غولومب رايس المدخلة مؤخرا إلى HEVC و إشارات المعاملات الترددية. في حين أنه في النهج الثاني، اخترنا كل الرموز المستخدمة لترميز المعاملات الترددية و كذا إشاراتها. في كلتا المقاربتين،وجدنا أن حجم و تركيبة الملف المشفر هما نفس نظيرتيهما للملف الغير المشفر. وبالتالي، فإن المساهمات المقدمة في هذه الأطروحة هي في الواقع خطوة أولى نحو أمن معيار HEVC.

كلمات مفتاحية : تشفير إنتقائي، ترميز الفيديو,HEVC, CABAC.

Table des matières

Liste des figures

Liste des tableaux

Chapitre I.

Introduction à l'information multimédia

I.1 Introduction

Les applications multimédias ont beaucoup émergées dans les dernières décennies dans de nombreux secteurs. En entreprise, celles-ci peuvent être la vidéophonie, la vidéoconférence. En éducation, on les trouve dans des applications de télé-éducation pour l'enseignement à distance. Et aussi, dans les lieux commerciaux, ces applications répondent aux besoins multiples comme l'achat (paiement a distance), et la publicité (messages publicitaires interactives en ligne).

Le terme *multimédia* est très récent. Si on fait une petite recherche linguistique, le terme *multimédia* est '' Ensemble des techniques et des produits qui permettent l'utilisation simultanée et interactive de plusieurs modes de représentation de l'information (textes, sons, images fixes, ou animées)'' selon le dictionnaire de Larousse [1]. Cependant, cette définition a besoin d'être élargie à d'autres types de données récemment inventées comme : la vidéo, l'hypertexte, et hypermédia. Multimedia est le pluriel de media ou medium qui est par définition '' Procédé permettant la distribution, la diffusion ou la communication d'œuvres, de documents, ou de messages sonores ou audiovisuels (presse, cinéma, affiche, radiodiffusion, télédiffusion, vidéographie, télédistribution, télématique, télécommunication) ''. Selon le vocabulaire d'ISO[1], la définition normalisée du terme media est ''moyen par lequel les données sont perçues, représentées, stockées ou transmises ''.

[1] **ISO/IEC 2382:2015 :** Technologies de l'information – Vocabulaire : http://www.iso.org/iso/fr/home/store/catalogue_tc/catalogue_detail.htm?csnumber=63598 [visité le 26/06/2015]

Apres son échantillonnage et sa numérisation, la représentation numérique au moyen de bits de l'information multimédia est effectuée avec ce qu'on appelle ''le codage'' en un format lisible par l'ordinateur. La représentation brute de l'information multimédia nécessite un grand espace de stockage et aussi une large bande passante pour sa transmission sur les réseaux. Le tableau I.1 montre les bandes passantes relatives à chaque type d'information multimédia. Pour pallier ce problème, on doit réduire au maximum la quantité de l'information multimédia émise/stockée en utilisant des techniques de compression.

	Texte	Image	Audio	Animation	Vidéo
Type d'objet multimédia	-ASCII EBCDIS	-image brute	-Flux binaire non encodé	-séquence d'images synchronisée avec une séquence audio	-séquence d'images a 25/30 images par secondes
Taille/bande passante	**≥2 KB par page**	**32 bits pour représenter chaque pixel**	**8 bits pour représenter 8 KHz.**	**2-5 MB/s pour 320×640×16 pixels par image**	**27 MB/s pour représenter 640×480×24 pixels/image**

Tableau I.1 Les différents types d'objets multimédias et les exigences taille/bande passante correspondantes.

Dans le reste de ce chapitre, on va découvrir un petit historique sur l'évolution de l'information multimédia. Après, on abordera la théorie de l'information pour se rapprocher au codage. Et on termine avec la compression.

I.2 Historique de l'évolution de l'information

Depuis son existence, la représentation et la communication de l'information ont constituée l'une des besoins indispensables pour l'être humain. Différentes types d'information ont été inventé a différentes époques temporelles comme il est illustré dans l'exemple de tableau I.2. Des gravures rupestres comme celles de Tassili en Algérie (voir figure I.1) donnent un exemple concret sur l'enregistrement de l'information daté avant de 5000 J.-C. Au moyen de l'alphabet, l'être humain a entré dans l'ère de l'écriture et de l'impression qui est caractérisé par la sauvegarde de l'information sur papier, et la transmission de savoirs a travers les livres. Le cumul de savoirs écrits a permis à l'être humain de se rendre à une nouvelle ère. C'est l'ère du monde d'informations analogiques et numériques qui a transformé le monde entier en

une petite ville accessible par des nouvelles media de communication comme la télévision numérique.

l'époque temporelle	Type de l'information	Les medium stockage	Les medium de transmission
Préhistoire : 15000 J.-C.	Communication à travers peintures, gravures, paroles, gestes,…	Les roches,…	
500 J.-C.	l'apparition des alphabets	Invention de papier	transmission des messages manuscrites a travers les animaux(les chevaux, pigeons,…)
400-1000	L'écriture	livres	L'apparition de premier système postal
1300-1800	Les news, magazines,…	Livres et librairies	L'imprimerie
1900	Code de morse, les signaux radio,…	La photographie, cinéma	Télégramme,
1950-1980	Télévision, téléphone,…	Les mémoires électroniques	Les satellites, diffusion radio et TV
1980-aujourd'hui	Les vidéos numériques avec leurs extensions,	Les disques durs, DVD-ROM,….	Ethernet, internet,

Tableau I.2 Illustration l'évolution de l'information a des époques différentes.

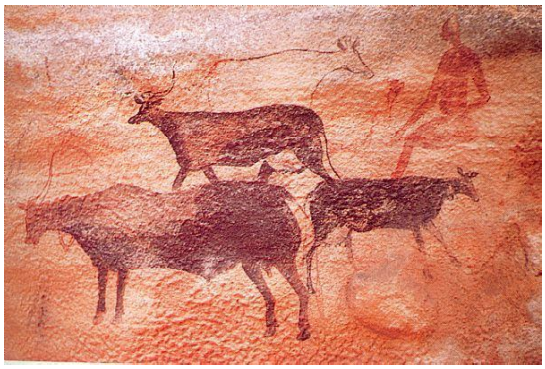


Figure I.1 Les gravures rupestres de Tassili [2].

I.3 Éléments de théorie de l'information et de codage entropique

Avant d'entrer dans le vif du sujet, le codage est un axe de recherche très important de la théorie de l'information qui remonte aux travaux de l'ingénieur américain Claude Elwood Shannon en 1948 [3] (figure I.2) dans le but de modéliser la communication sans/avec bruit en optimisant ainsi la transmission de flux d'information depuis une source émettrice jusqu'à un utilisateur. Ses travaux

permettent d'installer le premier pas vers l'ère de la communication numérique moderne.

Le codage permet d'établir une correspondance sans ambigüité et de passer d'une représentation externe d'une information (physique ou moral) vers une représentation interne suivant un ensemble de règles précises en utilisant un ensemble fini de symboles, par exemple, l'information huit est une représentation externe d'un nombre; par contre, 100 est une représentation interne de ordinateur.

Figure I.2 Claude Elwood Shannon.

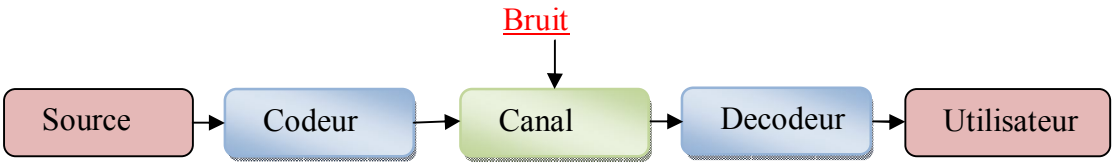


Figure I.3 une chaine de communication.

La figure I.3 ci-dessous illustre un système de communication entre une source émettrice et un utilisateur. Une source peut être tout signal (analogique ou numérique) capable de générer des informations comme la voix, signal radar, ou une séquence binaire. Un canal est une voie de transmission de l'information comme une ligne téléphonique, liaison radar. La transmission de l'information a travers un canal peut subir un bruit issu de caractéristiques physiques de canal lui-même, ou a cause des facteurs qui dépend de l'environnement. Le rôle principal de codeur est très varié, et il peut être une modulation pour adapter le signal émis aux caractéristiques du canal, une compression pour réduire la quantité d'information transmise, ou l'ajout de redondances pour combattre les dégradations de bruit afin de restituer le signal a la sortie. Finalement, le décodeur a pour rôle de reconstruire le signal a la sortie pour

que l'utilisateur puisse lire le message reçu. On peut considérer qu'un message est un événement aléatoire produit par la source avec un objectif est de lier la quantité d'information d'un message a sa probabilité d'émission. Ainsi, la quantité sera nulle si le message est certain c.-à-d. arrivé à l'utilisateur.

On représente pour la suite de cette section, l'alphabet d'une source comme un ensemble fini de N symboles$(S)_N = \{a_1, a_2, \dots, a_N\}$. Chaque symbole d'alphabet peut être une simple lettre ou un message. La source transmet chaque symbole de l'alphabet selon une loi de probabilité $(P)_N = (p(a_1), p(a_2), \dots, p(a_N))$, sachant que $\bigcup_i a_i$ représente l'événement certain et sur, c.-à-d. $\sum_i p(a_i) = 1$.

I.3.1 Les différents types de sources

Selon la probabilité d'émission de chaque symbole a_i, on distingue trois types de sources :

- une source *simple* ou sans mémoire si la probabilité d'émission d'une séquence de symboles $S_n = a_{t1}, a_{t2}, \dots, a_{tn}$ transmises a des instants successifs ti vérifie l'équation suivante :

$$p(a_{t1}, a_{t2}, \dots, a_{tn}) = p(a_{t1})p(a_{t2}) .. p(a_{tn}) \quad \text{(I.1)}$$

-une source avec *mémoire ou de Markov d'ordre r* si la probabilité d'émission de symbole a_t ne dépends que des *r* symboles précédents, c'est-à-dire :

$$p(a_t/a_{t1}, a_{t2}, \dots, a_{tn}) = p(a_t/a_{t-1}, a_{t-2}, \dots, a_{t-r}) \quad \text{(I.2)}$$

-une source est dit *stationnaire* si la probabilité d'émission de chaque symbole est indépendante de temps d'émission, autrement dit :

$$p(a_t) = p(a_{t+k}) \quad \text{(I.3)}$$

I.3.2 Entropie d'une source simple

Puisque la quantité d'information transmissible est proportionnelle à son degré d'incertitude, on peut définir l'entropie d'une source comme le nombre moyen minimal de bits par symbole nécessaire pour représenter la source. L'entropie d'une source simple S notée $H(S)$ peut être calculée comme suit :

$$H(S) = -\sum_{i=1}^{N} p(s_i) log_2(p(s_i)) \quad \text{(I.4)}$$

On note que l'entropie $H(S)$ est maximale si tous les symboles (ou messages) sont équiprobables. On rappelle aussi que chaque symbole s_i apporte une information de grandeur – $log(p(s_i))$.

I.3.3 Codage entropique

Le codage entropique est un schéma de compression statistique qui permet de réduire considérablement la quantifié d'information transmise par une source d'émission. Le but de codage est d'associer à chaque symbole de source un mot de code (codeword) qui sera représenté par une suite de bits. Deux types de codage peuvent être envisagés : le codage a longueur fixe et le codage a longueur variable. La distinction entre les mots de codes est nécessaire, et pour cela, on dit qu'un code est régulier s'il n'existe pas deux mots de même code, autrement dit, l'application de codage doit être injective.

I.3.3.1 Codage a longueur fixe

Un code de longueur fixe *FLC* (fixed length code) est un code dont tous les mots de codes ont la même longueur. Plusieurs représentations de codes sont possibles. Cependant, le codage optimal est celui qui utilise des codes réguliers de longueur n avec $log(N) < n < log(N) + 1$.

On note ici que si $H(S) = log(N)$, alors tous les symboles sont équiprobables.

I.3.3.2 Codage a longueur variable

Comme son nom l'indique, ce type de codage utilise des codes à longueur variable *VLC* (variable length codes) pour représenter les symboles. Cependant certain type de codage *VLC* assigne des codes plus court pour les symboles les plus fréquents, et des codes longs aux symboles les moins fréquents.

I.3.3.2.1 Le codage de Huffman

En 1952, David Albert Huffman a publié dans son article [4] une solution optimale pour le codage a longueur variable qui permet de réduire considérablement la quantité

d'information transmise via un canal non-bruité jusqu'à un taux de compression de 20-90% (si le canal est bruité, on ajoute des redondances ou des codes correcteurs d'erreurs pour combattre le bruit). Le principe général est d'associer des codes courts aux symboles les plus fréquents (c.-à-d. plus utilisés), et d'associer des codes longs aux symboles les moins fréquents (c.-à-d. rare). Grace à sa fiabilité, il est adopté par plusieurs logiciels de compression, et aussi, il est intégré en cascade dans plusieurs normes de compression audiovisuelle.

Le codage de Huffman consiste a construire un arbre binaire dont les feuilles terminales sont étiquetées par les symboles de l'alphabet en adoptant l'algorithme suivant :

Algorithme1 : Codage de Huffman.
Entrée : la séquence de symboles et ses probabilités respectives. **Sortie :** Arbre binaire.
1. ordonner les symboles par ordre croissant selon leurs probabilités d'occurrence. 2. Choisissez les deux symboles avec probabilités les plus faibles et les fusionner en un nouveau symbole auxiliaire. 3. Calculer la probabilité de ce nouveau symbole auxiliaire. 4. Si plus d'un symbole reste, répétez les étapes 2 et 3 pour le nouvel alphabet auxiliaire. 5. ordonner les symboles par ordre croissant selon leurs probabilités d'occurrence. 6. Choisissez les deux symboles avec probabilités les plus faibles et les fusionner en un nouveau symbole auxiliaire. 7. Calculer la probabilité de ce nouveau symbole auxiliaire.

On forme le code de mot en parcourant l'arbre ainsi créé en allant de la racine (symbole de probabilité égale à 1) vers les feuilles (symboles de départ). On attribue 0 aux branches gauches et 1 aux branches droites.

I.3.3.2.2 Le codage arithmétique

Ce type de codage [4] permet d'associer un nombre appartenant à l'intervalle [0,1] à une chaine de symboles. Malgré son implémentation logicielle est très délicate et couteuse en temps de calcul, il adopté par les normes de compression vidéo récentes grâce a son efficacité comparant par rapport au codage de Huffman.

Le codage arithmétique peut être résumé par l'algorithme suivant :

Algorithme2 : Codage arithmétique.

Entrée : la séquence de symboles et ses probabilités respectives.

Sortie : $[low, high[$

1. Calculer la probabilité associée à chaque symbole dans la chaîne à coder.
2. Associer à chaque symbole un sous-intervalle de [0,1[proportionnel à sa probabilité (l'ordre de rangement des intervalles sera mémorisé car il est nécessaire au décodeur, et pour cela on utilise la probabilité cumulée).
3. Tant qu'il reste un symbole dans la chaîne à coder :

(a) range = high – low

(b) low = low + range ×(limite basse du sous intervalle du symbole)

(c) high = low + range× (limite haute du sous intervalle du symbole)

4. Choisir un nombre $\in [low, high[$ pour coder la séquence.

La figure I.4 montre le processus de codage de la séquence CAEE, $ est le caractère de fin de cette séquence. On remarque que le choix de 0.331 est suffisant pour coder cette séquence.

I.4 La compression

Si le codage entropique permet de représenter en termes de bits les symboles d'une source, la compression permet en outre de réduire le flux de bits (ou le débit) émis via le canal. En effet, la compression est un algorithme qui permet de réduire la taille du flux de bits de la donnée brute transmise depuis la source jusqu'a l'utilisateur. Autrement dit, la compression est une application qui permet le passage d'une représentation brute R d'une donnée numérique vers une autre représentation dites comprimé C dont la taille est en général inferieur a celle d'origine. La décompression est la reconstruction de R à partir de C.

En général, un algorithme de compression se procède en deux étapes : l'extraction de l'information pertinente et l'élimination de redondance. La détermination de l'information pertinente dépend toujours de la nature de donnée brute à comprimer. Plusieurs techniques [6] peuvent achever l'extraction de l'information pertinente comme la quantification des coefficients pour les données audiovisuelles. L'élimination de redondance peut être atteindre par des approches prédictives, des

approches basées sur des transformées, ou par un codage entropique comme celui de Huffman.

Symbole	probabilité	Range
A	0.2	[0,0.2[
B	0.1	[0.2,0.3[
C	0.2	[0.3,0.5[
D	0.05	[0.5,0.55[
E	0.3	[0.55,0.85[
F	0.05	[0.85,0.9[
$	0.1	[0.9,1[

(a)

Symbole	Low	High	range
	0	1.0	1.0
C	0.3	0.5	0.2
A	0.30	0.34	0.04
E	0.322	0.334	0.012
E	0.3286	03322	0.0036
$	0.33184	0.33220	0.00036

(b)

Figure I.4 Exemple d'un processus de codage arithmétique : (a) les probabilités associées aux symboles A,B,C,D,E,F,$, (b) Codage de la séquence CAEE$.

La qualité de la donnée reconstruite *R'* après la décompression de la donnée comprimée C est un facteur très important pour l'évaluation de l'algorithme de compression. En effet, on dit qu'un algorithme de compression est sans perte si la donnée comprimée est identique à l'originale. Réciproquement, on qualifie un algorithme de compression avec perte si une dégradation affecte la donnée reconstruite *R'* après la compression de *R*. Cette dégradation est appelée souvent la distorsion ou la perte de qualité.

L'efficacité d'un algorithme de compression peut être évaluée en utilisant les quantités suivantes :

a) rapport de compression (taux) : il correspond au rapport $\frac{la\ taille\ de\ C}{la\ taille\ de\ R}$, et il est généralement inferieur à 1.

b) le facteur de compression : est le rapport inverse de rapport de compression, c.-à-d., plus la compression est fiable, plus le facteur croit.

c) taux en pourcentage est souvent utilisé, et il est calculé avec la formule$(1 - rapport\ de\ compression) \times 100$.

I.4.1 Compression sans perte (lossless compression)

La compression sans perte est un algorithme dont l'information reconstruite R' est identique à celle d'origine R. Des données comme le texte ou les images médicales exigent qu'il n'ait pas une perte de l'information après la compression, et ce type d'algorithme est très approprié car il peut garder l'information originale sans distorsion. Par conséquent, ce type d'algorithmes considère l'information originale comme une information pertinente, et ils sont restreint a éliminer seulement les redondances statistiques observée.

Les algorithmes de compression sans perte peuvent réduire le débit d'une source R jusqu'a $H(R)$ bits, et ils ne peuvent pas atteindre un débit moins inferieurs car ils vont engendrer une distorsion significative. Ainsi, on qualifie la redondance par $H(R) - H(C)$ bits.

Les algorithmes de compression sans perte sont trop nombreux et le lecteur pourra recourir à [6] pour une documentation approfondi. Parmi eux, on peut citer à titre d'exemple :

a) le codage de Huffman.
b) le codage arithmétique.
c) le codage *RLE* (run-length encoding) ou codage par plage : est utilisé par de nombreux formats d'images (BMP, TIFF,…). Il est basé sur le codage de nombre de répétions consécutives. Une première valeur donne le nombre de répétitions, et une seconde valeur donne le symbole répété.
d) le codage *LZW* (Lempel-Ziv-Welch) : est un algorithme inventé par Abraham Lempel, Jacob Ziv, et Terry Welch et est utilisé par plusieurs applications comme le compresseur WinZip. LZW est algorithme très rapide aussi bien en décompression qu'en compression. Son principe est de repérer des séquences qui apparaissent plusieurs fois, en construisant au fur et à mesure un dictionnaire de séquences, et de les remplacer par leurs indices dans le dictionnaire. Le dictionnaire fait partie intégrante des données compressées. Pour que cette méthode soit efficace, il ne faut donc pas que sa taille soit supérieure à celle économisée en recodant les données.

I.4.2 Compression avec perte (lossy compression)

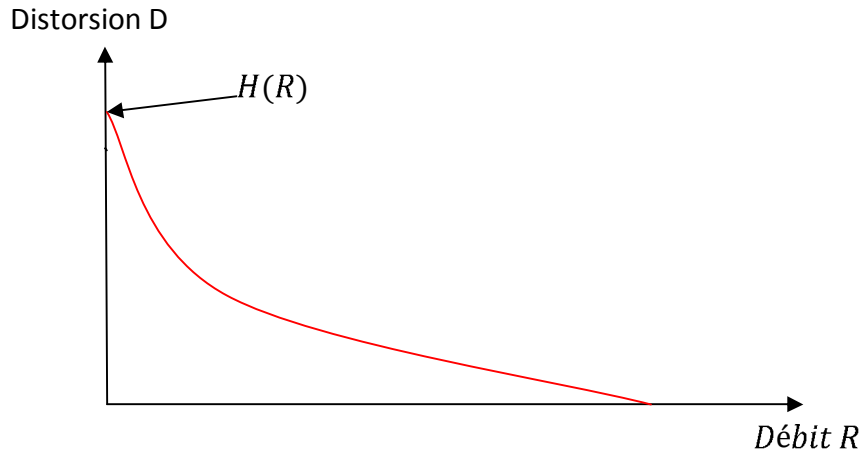


Figure I.5 La courbe débit-distorsion $R(D)$.

Comme son nom l'indique, la compression avec perte résulte une perte de qualité après la réduction du débit. La figure I.5 illustre la variation de distorsion en fonction du débit (nombre de ***bits/symbole***) au moyen de la courbe $R(D)$. La compression optimale avec une distorsion nulle est possible quand le débit sera égal a $H(R)$. Par contre, la distorsion augmente avec la décroissance du débit. Par conséquent, l'algorithme de compression devra tenir en compte la minimisation du compromis débit-distorsion. Et cela est possible avec des techniques d'optimisation mathématique.

Une très grande variété de techniques de compression avec perte existent [6] pour tout type de media, et on va restreindre ici a quelques approches populaires en commençant tout d'abord avec les mesures de calcul de distorsion, après, on entame a la quantification, les transformées, et des approches de quantification prédictive.

I.4.3 Calcul de la distorsion

Soit deux signaux numériques $x = \{x(i), i = 1, \dots, n\}$ et $y = \{y(i), i = 1, \dots, n\}$, avec x et y représentent le signal brut et le signal comprimé respectivement, et n représente le nombre d'échantillons pour chaque signal. Alors pour évaluer la distorsion D entre x et y, on doit utiliser soit des mesures subjectives qui nécessitent la présence d'un expert spécialisé dans le domaine (comme en imagerie médicale ou

satellitaire), ou bien par l'utilisation des mesures objectives qui s'appuient sur des calculs par des formules mathématiques comme l'erreur quadratique moyen (noté *MSE* pour mean square error) qui peut être calculé avec la formule suivante :

$$MSE = \frac{1}{n}\sqrt{\sum_{i=1}^{n}|x(i) - y(i)|^2} \quad \text{(I.4)}$$

I.4.4 La quantification

La quantification est le procédé permettant de représenter le signal en un ensemble *fini* et *réduit* de symboles discrets afin de simplifier la représentation du signal. Cette réduction permet en conséquence de minimiser aussi bien le débit qu'en l'entropie. La quantification dont une seule valeur d'entrée est scalaire est appelée la quantification scalaire, par contre la quantification vectorielle cherche à associer un ensemble de vecteurs a un vecteur quantifié.

I.4.4.1 La quantification scalaire

L'objectif de la quantification scalaire est de deviser la dynamique de la grandeur physique du signal (l'intervalle dont les extrémités sont la minimale et la maximale valeur de l'amplitude de signal) en un nombre fini d'intervalles, et d'associer a chaque intervalle une seule valeur. Ici on distingue entre deux types de quantification scalaire : la quantification uniforme dont les longueurs des intervalles sont égales, et la quantification non uniforme qui permet de choisir la longueur de chaque intervalle suivant la loi de distribution de probabilité de chaque échantillon du signal.

I.4.4.2 La quantification vectorielle

La quantification vectoriel peut être vu comme une application associant un élément d'un ensemble de vecteurs a un seul vecteur représentant une sortie choisie parmi un dictionnaire (appelé aussi code-book). Elle dépend dans sa conception de plusieurs paramètres : le choix de l'entité vectorielle, mesure de distorsion, génération et organisation de dictionnaire,…

Les algorithmes d'apprentissage comme les réseaux de neurones [8], kppv [9] (k-plus proches voisins) sont des approches prometteuses pour la quantification vectorielle surtout si le dictionnaire est défini auparavant. Cependant, on trouve des

algorithmes comme l'algorithme de LBG [10] qui permet de générer le code-book efficacement en commençant avec un code-book initial.

I.4.5 Le codage prédictif

La notion de redondance est liée à celle de prédictibilité. Lorsque le signal est redondant, il est possible de prédire un échantillon à partir des échantillons passés. L'idée du codage prédictif est alors à ne pas quantifier et à coder que la partie non prédictible du signal qui représente souvent l'erreur résiduelle de prédiction ou le résidu, et qui est calculé comme suit :

$$e(n) = x(n) - \hat{x}(n) \tag{I.5}$$

avec x(n), $\hat{x}(n)$, et e(n) représentent respectivement le signal source, le signal prédit et l'erreur résiduelle. La dernière sera généralement quantifiée, codée, et transmise au décodeur.

I.4.6 Le codage par transformée

Supposons que l'on ait un bloc d'échantillons successifs (vecteur) d'un processus aléatoire stationnaire à coder avec un nombre fixe de bits. Appelons X ce vecteur aléatoire avec $X = (X_1, X_2, \dots, X_N)^T$. Ces échantillons qui d'après l'hypothèse de stationnarité ont la même variance peuvent présenter une corrélation importante. Cette corrélation entraîne une certaine redondance qui est conservée dans les échantillons quantifiés. L'idée du codage par transformée est qu'en opérant une transformation linéaire sur X, on peut obtenir un nouveau vecteur Y dont les composantes sont moins corrélées que celles de X, et que l'information peut y être plus compacte, c'est-à-dire concentrée sur quelques composantes, au lieu d'être uniformément répartie sur toutes les composantes. On espère alors quantifier ces composantes de façon plus efficace que pour X.

Il y a aussi une raison ‘’subjective’’ pour utiliser une transformée qui est de se référer aux outils perceptifs humains qui interviennent pour la vue ou d'audition. L'oreille en particulier, opère au niveau de la membrane basilaire (dans l'oreille interne) une transformation du signal temporel acoustique en influx nerveux répartis suivant une échelle fréquentielle sur les fibres nerveuses du nerf auditif. Ces fibres se

comportent, en première approximation et partiellement, comme un banc de filtres dont les fréquences centrales sont disposées sur une échelle pseudo-logarithmique. Quant a la vue, l'œil humain est moins sensible aux hautes fréquences qu'aux hautes fréquences. Par Conséquence, il s'avère que le passage du domaine spatiale vers le domaine fréquentiel en utilisant des transformées mathématiques orthogonales inversible et linéaire est très fiable pour l'extraction de l'information pertinente et l'élimination d'une redondance dans le signal.

Parmi les transformations qui sont couramment adoptée pour la compression, on peut citer : la transformation de Karhunen-Loeve (*TKL*) [11], la transformation de Fourrier discrète (*TFD*) [12], la transformation de Hadamard (*TH*) [13], la transformation par ondelettes [14], et la transformation en cosinus discrète (DCT pour discret cosinus transform) [15]. Cette dernière est la plus populaire car les coefficients sont décolérés, et l'énergie du signal est concentrée sur un ensemble déterminé de coefficients. En plus, elle est utilisée dans la majorité des normes récentes de compression visuelle comme JPEG [16], MPEG[17][18][19], et H.264 [20]. Dans la suite, on va décrire l'application bidimensionnelle de cette transformée et les avantages prometteuses en compression qu'elle offre.

I.4.6.1 La transformée en cosinus discrète

Cette transformée est un cas particulier de transformée de Fourier discrète car elle donne des coefficients réels contrairement au TFD qui donner des coefficients complexes. Sa première version unidimensionnelle inventée en 1974 par Ahmed, Natarajan et Rao [15] est donnée par la formule suivante :

$$\begin{cases} Y(0) = \frac{\sqrt{2}}{N}\sum_{n=0}^{N-1} x(n) \\ Y(k) = \frac{2}{N}\sum_{n=0}^{N-1} x(n).\cos\left(\frac{k.\pi.(2n+1)}{2n}\right), avec\ k = 1,2,\dots,N-1 \end{cases} \quad (I.6)$$

avec $x = \{x(n), n = 0,\dots,N-1\}$, et Y représente la représentation de x en espace fréquentiel. La transformée en cosinus discrete inverse s'obtient avec :

$x(n) = \sum_{k=0}^{N-1} C(k)Y(k)\left(\frac{k.\pi.(2n+1)}{2n}\right), avec\ n = 0,1,\dots,N-1$, et $C(0) = \sqrt{\frac{1}{N}}$ et $C(k) = \sqrt{\frac{2}{N}}$ pour k=1,…,N-1.

En 1978, une version bidimensionnelle [21] de la DCT est publiée, et qui peut être calculée comme suit :

$$F(u,v) = \frac{4C(u)C(v)}{x^2}\sum_{j=0}^{n-1}\sum_{k=0}^{n-1} f(i,j)\cos\left(\frac{(2j+1)u\pi}{2n}\right)\cos\left(\frac{(2k+1)v\pi}{2n}\right) \quad \text{(I.7)}$$

avec $f = \{f(i,j),(i,j) \in \{0,\dots,n-1\}^2\}$ represnete un signal bidimensionnelle, et $F = \{F(u,v),(x,y) \in \{0,\dots,n-1\}^2\}$ represente la représentation fréquentielle de *f*.

et aussi $c(w) = \begin{cases} \frac{1}{2}, si\ w = 0 \\ 1, si\ w = 1,2,\dots,n-1 \end{cases}$

La transformée en cosinus discrète inverse est donnée comme suit :

$$f(j,k) = \sum_{u=0}^{n-1}\sum_{k=0}^{n-1} c(u)c(v)F(u,v)\cos\left(\frac{(2j+1)u\pi}{2n}\right)\cos\left(\frac{(2k+1)v\pi}{2n}\right) \quad \text{(I.8)}$$

La transformée DCT a une autre forme matricielle telle que :

$$\begin{cases} F = AfA^T \\ f = A^TFA \end{cases} \quad \text{(I.9)}$$

avec $A_{ij} = C_i\left(\frac{(2j+1)i\pi}{2n}\right)$.

Cette forme matricielle est appropriée pour les applications visuelles car elle est très rapide en temps de calcul.

Par exemple, pour n=4, alors

$$A = \begin{pmatrix} \frac{1}{2}\cos(0) & \frac{1}{2}\cos(0) & \frac{1}{2}\cos(0) & \frac{1}{2}\cos(0) \\ \sqrt{\frac{1}{2}}\cos\left(\frac{\pi}{8}\right) & \sqrt{\frac{1}{2}}\cos\left(\frac{3\pi}{8}\right) & \sqrt{\frac{1}{2}}\cos\left(\frac{5\pi}{8}\right) & \sqrt{\frac{1}{2}}\cos\left(7\frac{\pi}{8}\right) \\ \sqrt{\frac{1}{2}}\cos\left(\frac{2\pi}{8}\right) & \sqrt{\frac{1}{2}}\cos\left(\frac{6\pi}{8}\right) & \sqrt{\frac{1}{2}}\cos\left(\frac{10\pi}{8}\right) & \sqrt{\frac{1}{2}}\cos\left(\frac{14\pi}{8}\right) \\ \sqrt{\frac{1}{2}}\cos\left(\frac{3\pi}{8}\right) & \sqrt{\frac{1}{2}}\cos\left(\frac{9\pi}{8}\right) & \sqrt{\frac{1}{2}}\cos\left(\frac{15\pi}{8}\right) & \sqrt{\frac{1}{2}}\cos\left(\frac{21\pi}{8}\right) \end{pmatrix} \quad \text{(I.10)}$$

Le coefficient d'indice (0,0) est appelé le coefficient DC et il représente l'énergie moyenne du signal, alors que les autres coefficients sont appelés les coefficients ACs. Ces derniers peuvent être distribués selon leur ordre en trois catégories : les coefficients à basse fréquence, les coefficients à fréquence moyenne, les coefficients à haute fréquence comme il est montré dans la figure I.6. Et aussi chacun d'eux représente une information particulière selon leur position dans la matrice des coefficients. La figure I.7 illustre les significations spatiales relatives aux coefficients

dans le cas où le signal est une image, par exemple, les coefficients diagonaux représentent les contours diagonaux dans l'image.

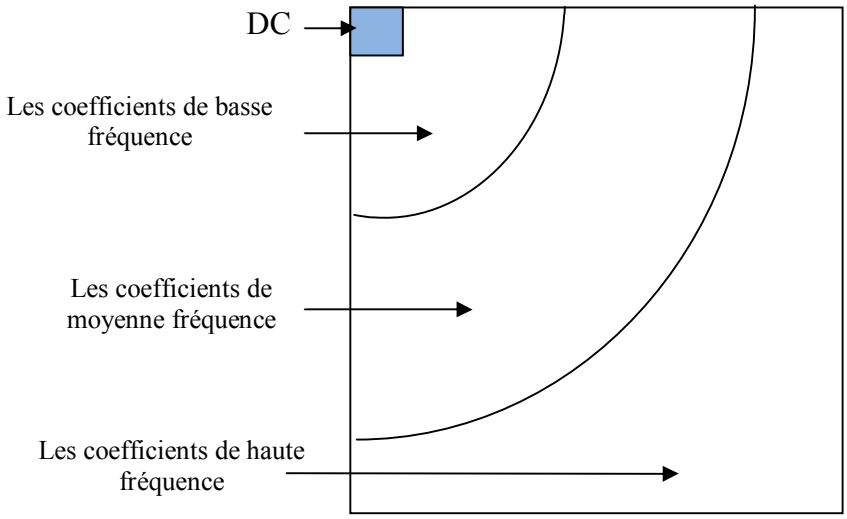


Figure I.6 Les différentes classes de coefficients selon leurs fréquences.

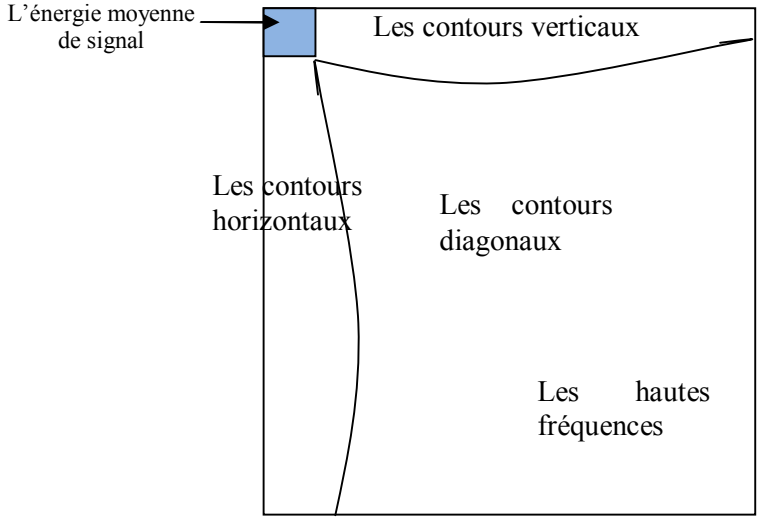


Figure I.7 Significations des coefficients de DCT.

L'objectif essentiel par le codage de transformée est de localiser les hautes fréquences afin de les éliminer au moyen de la quantification. Cette élimination engendre des coefficients quantifiés nulles dans la partie inferieure droite, ce qui permet de faciliter le codage entropique par la suite.

I.5 Conclusion

Dans ce chapitre, nous avons commencé par présenter l'évolution de l'information multimédia où l'être humain a tenté toujours d'améliorer ses moyens de communications et ses solutions proposées pour transmettre ses informations.

Nous avons présenté aussi une petite introduction sur le codage entropique qui se devise en codage à longueur fixe et variable. Le codage entropique permet de coder la source en assignant des codes binaires aux symboles générés. Le codage à longueur variable est plus préféré car il permet d'éliminer statistiquement les redondances observées au sein de l'information. La compression est aussi abordée dans ce chapitre sous ses deux types sans ou avec perte.

Dans le prochain chapitre, nous allons aborder la compression des signaux visuels images et vidéo, où elle repose sur la mise en application en cascade de tous les outils de compression sans/avec perte précédemment discutés vus dans le présent chapitre.

Chapitre II.

La compression de la vidéo : de notions de base à la norme HEVC

II.1 Introduction

L'information vidéo est l'une des formats les plus utilisées aujourd'hui, et elle occupe une place dominante grâce à son importance. On la trouve partout dans des applications diverses : la vidéo conférence, la vidéo surveillance, la télévision,…etc.

Puisque la vidéo n'est qu'un défilement d'une séquence d'images animées, nous allons commencer par jeter un bref coup d'œil sur la compression d'images numériques. Après, nous allons passer à la compression de l'information vidéo en expliquant les diverses étapes de compression liées. Aussi, on va citer quelques normes de compression vidéo normalisées depuis les deux groupes de normalisation IEC/ISO et ITU-T[2]. Finalement, on terminera avec une conclusion afin de clôturer le présent chapitre.

II.2 L'image numérique

Avec l'avènement de l'ère numérique, l'image numérique (digital image) a remplacé son prédécesseur analogique dans les champs de recherche et d'applications, ce qui permet en revanche de faciliter la sauvegarde, le codage et la transmission de celle-ci afin de la déployer dans des applications variées de traitement d'images comme la segmentation, ou en reconnaissance de formes comme la reconnaissance de l'empreinte digitale.

[2] "ITU-T Recommendations," *ITU*. [Online]. Available: http://www.itu.int/en/ITU-T/publications/Pages/recs.aspx. [Accessed: 29-May-2015].

L'image numérique est tout simplement, un signal numérique bidimensionnel qui prend la forme d'une grille rectangulaire comme le montre la figure II.1. Cette dernière comporte plusieurs points appelés pixels (Picture Element). Chacun d'eux permet de représenter la valeur d'une couleur échantillonnée et est identifié par sa position dans l'image. Le nombre de pixels représentant l'image donne la définition de celle-ci, et il est calculé comme le nombre de pixels dans une seule ligne multiplié par le nombre de pixels dans la colonne ; on la note brièvement par $M \times L$ ou L et M représente le nombre de ligne et de colonne respectivement. La résolution d'une image est le nombre de pixels par pouce (une pouce=2.56 cm), et elle est exprimé en points par pouce **PPP**. Cependant, le terme résolution remplace souvent celui de la définition dans beaucoup de littérature scientifique.

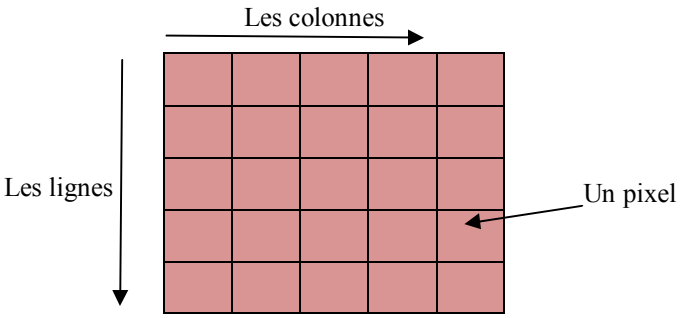


Figure II.1un exemple matriciel d'une image numérique.

Figure II.2 La première image numérique capturée en 1957.

Il est d'usage aussi de connaitre que la première image numérique est acquise en 1957 pour un enfant de trois mois (voir la figure II.2) dont sa définition est 157×157 pixels.

II.3 Les espaces de couleurs

La couleur en traitement d'image est représentée en un espace vectoriel orthogonal dont chaque composante donne une information pertinente sur la couleur. En conséquence, une image peut être codée en utilisant plusieurs matrices dont chacune représente une composante dans l'espace choisi. L'espace de couleur le plus répandu est celui de RGB (Red, Green, Blue). Chaque vecteur correspond à une couleur spécifique suivant deux synthèses différentes : additives ou soustractives (voir figure II.3). Aussi, chaque composante est codée souvent sur huit bits ce qui permet de coder $2^{8+8+8} = 2^{24}$ couleurs possibles. Les vecteurs $(0,0,0)^T$ et $(255,255,255)^T$ reflètent les couleurs de noir et de blanc en synthèse additive respectivement.

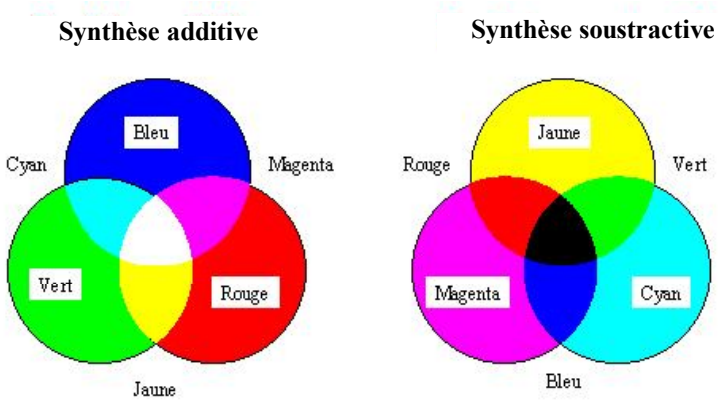


Figure II.3 Les deux synthèses de couleur utilisées en espace RGB.

Une des inconvénients majeurs du système RGB est que les composantes sont très corrélées. La commission Internationale de l'Eclairage[3] a défini plusieurs espaces colorimétriques parmi lesquelles on trouve le système YCbCr (Y pour luminance, Cb et Cr pour les chrominances bleu et rouge respectivement) qui est très adopté dans les applications de compression d'images et de vidéo.

Ces systèmes exploitent le fait que le cerveau traduit le signal trichromatique perçu par l'œil, comme un signal composé de trois composantes, dont l'une est achromatique : la luminance. Elle permet d'éclaircir ou d'assombrir une couleur en ajustant la quantité de noir. L'information de couleur peut être représentée en utilisant

[3] http://www.cie.co.at/

la chrominance (la différence de chaque couleur par rapport à la luminance). La figure II.4 montre une image avec ses composantes respectives en systèmes YCbCr

Le passage de système RGB vers le système YCbCr est obtenu par :

$$\begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.144 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{pmatrix} \times \begin{pmatrix} R \\ G \\ B \end{pmatrix} + \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} \quad \text{(II.1)}$$

avec $(Y, Cb, Cr)^T$et $(R, G, B)^T$represente les vecteurs YCbCr et RGB respectivement. La luminance Y est calculée en utilisant une somme pondéré entre les composantes rouge, vert et bleu, et elle représente souvent le niveau de gris de l'image. Le passage de système YCbCr vers le système RGB s'obtient avec :

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1.402 \\ 1 & -0.344 & -0.714 \\ 1 & 1.722 & 0 \end{pmatrix} \times \begin{pmatrix} Y \\ Cb \\ Cr \end{pmatrix} - \begin{pmatrix} 0 \\ 128 \\ 128 \end{pmatrix} \quad \text{(II.2)}$$

Figure II.4 une image avec ses composantes respectives en systèmes YCbCr : (a) l'image originale, (b) la composante Y, (c) La composante Cb, (d) la composante Cr.

II.4 Les formats de sous-échantillonnage de chrominance

Le système visuel humain est toujours sensible aux variations de luminances que celles de chrominances de couleur. Cette imperfection visuelle de l'œil est toujours tenue en compte lors de l'échantillonnage et de la numérisation de couleur, où on sous-échantillonne seulement les chrominances Cr et Cb respectivement. Pour cela, plusieurs formats sont définis. Les plus populaire sont 4:4:4, 4 :2:2, et 4:2:0, est sont montré dans la figure II.5.

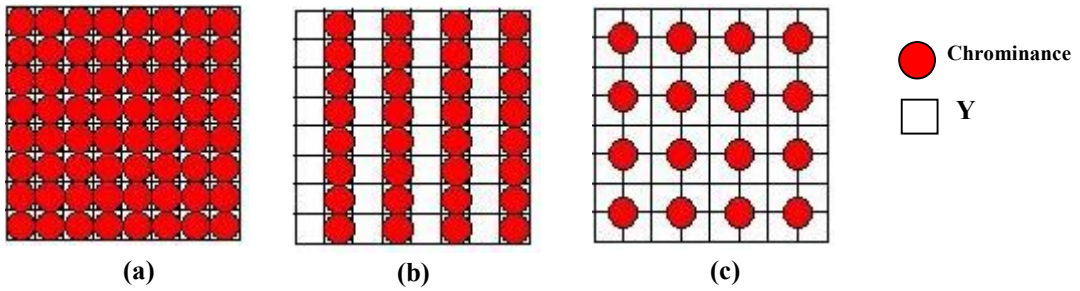


Figure II.5 Les formats de sous-échantillonnage de chrominance : (a)4 :4 :4, (b) 4 :2 :2, et (c) 4 :2 :0

Le format 4:4:4 explique qu'il n'y a pas de sous-échantillonnage de chrominance, autrement dit, c'est un échantillonnage avec haute fidélité. Chaque échantillon de luminance possède son correspondant pour les chrominances Cr et Cb. Cependant, cet échantillonnage parfait n'est pas le même pour 4:2:2 et 4:2:0.

Le format 4:2:2 signifie que pour chaque quatre échantillons de luminance dans la direction horizontale, il y aura deux échantillons pour Cr et deux autres pour Cb.

Finalement, le format 4:2:0 signifie que pour chaque groupe rectangulaire formé de quatre échantillons de luminance, on trouve seulement un seul échantillon de chrominance Cr, et aussi un seul pour Cb.

II.5 Introduction a la compression d'images fixes

Apres son acquisition et son numérisation par des capteurs numériques, une image numérique au format brute exige un espace de stockage très important pour sa sauvegarde. Elle correspond à une quantité d'information significative qui s'accroit en fonction de la résolution de l'image, espace de couleur choisi, et en nombre de bits requis pour coder chaque pixel. Par conséquent, la compression sans ou avec perte s'avère nécessaire pour réduire la taille de cette image brute.

On trouve pour la compression d'images sans perte plusieurs formats qui se distinguent selon le niveau de compression souhaité et les applications visées pour leur création. La qualité d'image reconstruite est identique à celle d'image brute. Parmi ces formats, on peut citer le format GIF (Graphics Interchange Format) [22] qui est inventé en 1987 afin de faciliter la transmission d'image aux utilisateurs. On

trouve aussi, le format PNG (Portable Network Graphics) [23] qui est approprié pour la sauvegarde d'images synthétisées comme les icones.

Les techniques de compression d'images avec perte permettent d'éliminer la redondance spatiale dans l'image en exploitant la corrélation qui existe entre un voisinage de chaque pixel. La compression avec perte opère une analyse fréquentielle locale de l'image. Les pertes engendrées sont contrôlées de manière à ce qu'elles restent invisibles à l'utilisateur.

Les travaux de comité d'expert JPEG (acronyme de *Joint Photographic Expert Group*) spécialisé dans la définition des formats compressés pour les images fixes ont permis de réaliser des techniques de compression d'images avec haute efficacité a savoir les formats JPEG [16][23] et JPEG2000 [25].

La norme JPEG est un standard qui permet de compresser des images monochromes (par exemple en niveaux de gris) ou des images couleurs. Selon le type de compression souhaité, cette norme incorpore deux modes de compression : JPEG-LS [23] pour la compression sans perte, et JPEG [16] pour la compression avec perte.

Le codeur JPEG-LS consiste à réduire la taille de l'image en transformant l'image brute en un flux binaire (bitstream) lisible lors du décodage. Il procède en deux étapes essentielles comme les montre la figure II.6 :

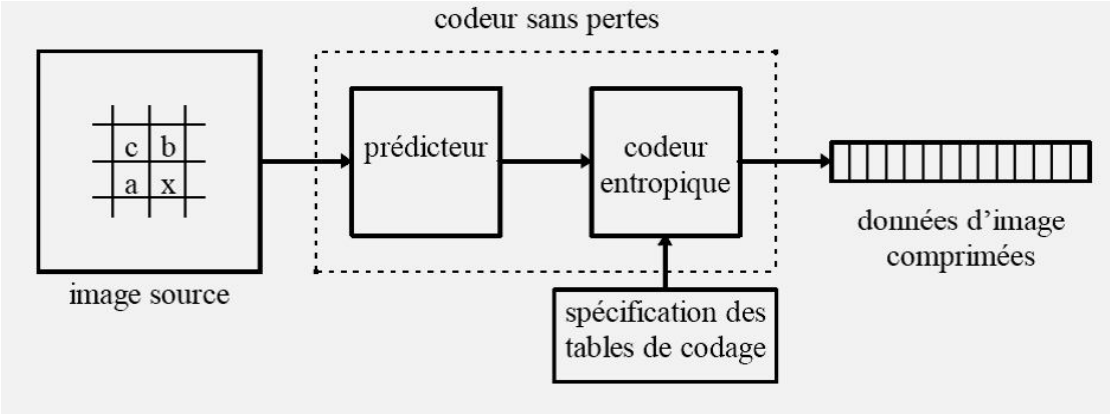


Figure II.6 Le schéma de compression adopté en JPEG-LS.

1) La prédiction : elle permet de choisir un prédicateur optimum $\hat{x}$ pour chaque pixel à coder x a partir de trois échantillons a, b, et c.

2) le codage entropique : l'erreur résiduelle $x - \hat{x}$ est ensuite codée par un codage typique de type Huffman ou de type arithmétique.

Le mode avec perte de JPEG (connu aussi sous le nom JPEG) standardisé en 1992 par une équipe commune de ISO et ITU-T sous la norme ISO/CEI 10918-1 UIT-T Recommendation T.81, et il supporte plusieurs modes opérationnels à savoir : le mode séquentiel qui est le mode par défaut, et les modes progressif et hiérarchique qui apporte des modifications de l'affichage lors de décodage pour accélérer la visualisation de l'image reconstruite.

La figure II.7 illustre parfaitement la compression d'une image en mode séquentiel. Premièrement, l'image à compresser est mise en forme par un découpage en bloc ; chaque composante couleur (luminance et chrominances) est découpé en bloc de 8×8 échantillons. Chaque bloc b_{ij} est soumis à une transformation fréquentielle en cosinus discrète. Comme les amplitudes des coefficients augmentent avec la diminution des fréquences (u, v), JPEG applique une quantification matricielle afin d'éliminer les hautes fréquences car leur absences est imperceptible par le système de visuel humain. De ce fait, on utilise l'équation suivante :

$$QTC_{uv} = round(B_{uv}/S_{uv}) \qquad \text{(II.3)}$$

avec round est une fonction qui arrondit l'argument d'entrée au entier le plus proche, S est la matrice de quantification définie dont chaque composante a sa propre matrice et QTC_{uv} représente le coefficient quantifié (Quantized transform coefficient) d'indice (u,v).

Le codage de DCs diffère de celui de ACs. Le coefficient DC est codé en mode prédictif suivant le principe de DPCM (acronyme de Differential pulse-code modulation). Les ACs quantifiés sont rangés dans une liste selon un parcours en zigzag qui permet de regrouper les ACs suivant leurs fréquences (voir la figure II.8). Ensuite, Lorsque tous les coefficients non nuls sont rangés dans la liste, le parcours s'arrête et le marqueur EOB (pour End Of Block) vient clôturer la liste. Un codage de type RLE est appliqué à cette liste pour éviter codage de valeurs répétées d'amplitudes des ACs. Finalement, les deux listes de QTCs (RLE et DPCM) sont soumises à une étape de codage entropique.

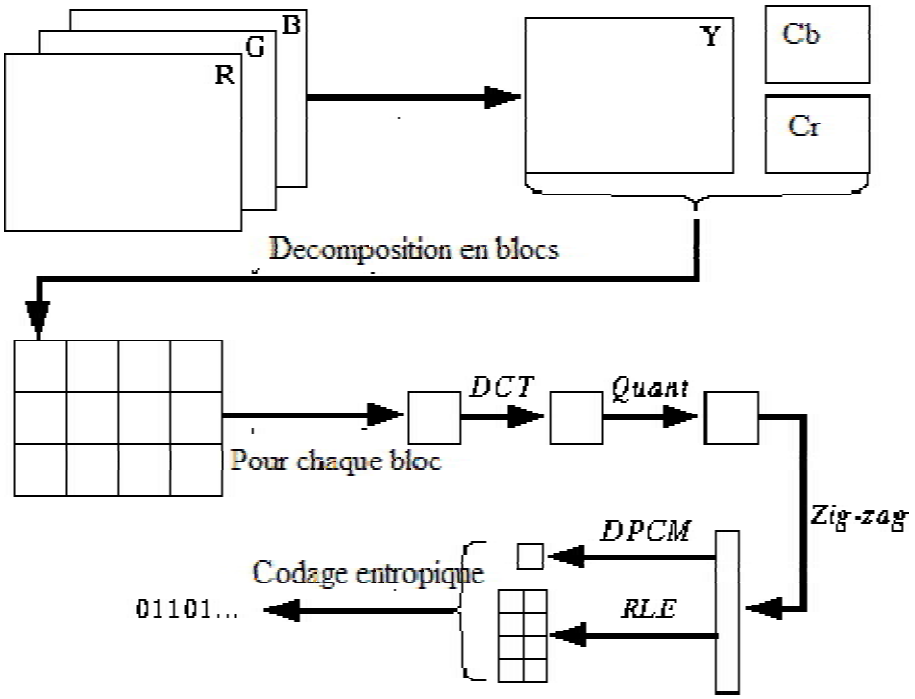


Figure II.7 Schéma de compression adopté en compression JPEG.

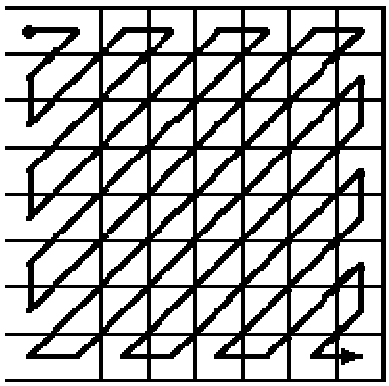


Figure II.8 Le parcours en ZigZag.

II.6 La vidéo numérique

La vidéo numérique est une séquence animée d'images fixes qui permet de représenter la même information ou la même scène temporellement (Voir figure II.9). Chaque séquence vidéo est caractérisée par des propriétés spatiales qui décrivent chaque image comme la résolution, l'espace de couleur choisi, le format de sous échantillonnage de couleur (pour le système YCrCb), et la résolution temporelle qui indique le nombre d'images par seconde *fps* (Frame per second). Chaque image est

nommée une trame (frame). Une dernière caractéristique est le débit (Bitrate) qui représente le nombre de bits nécessaire pour la transmission par seconde.

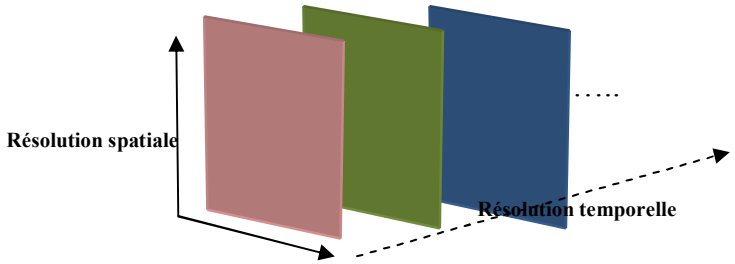


Figure II.9 Les résolutions spatiale et temporelle d'une séquence vidéo.

Chaque image est affichée de haut en bas, et de gauche à droite selon un mode de balayage (scan). Quant à la télévision, les images sont affichées entrelacée ; les lignes impaires (top field) de la première image suivie par les lignes paires (bottom field) de sa suivantes comme le montre la figure II.10. A contrario, le mode progressif qui est beaucoup employé dans des applications informatiques, affiche chaque trame entièrement. Le mode progressif est noté ''p'', par contre le mode entrelacé est noté ''i''.

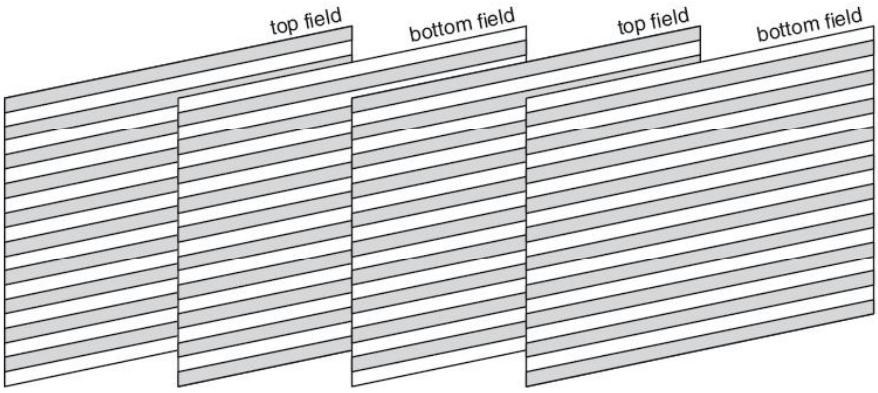


Figure II.10 Le mode d'affichage entrelacé.

II.7 Les formats populaires de la vidéo numérique

La diversification d'applications numériques qui utilise la vidéo, et la révolution .technologique visuelle et réseau permettent l'émergence de plusieurs formats vidéo qui varient selon la résolution, nombre d'image par seconde, et le mode de balayage utilisé lors de l'affichage (voir le tableau II.1). Les applications mobiles utilisent

généralement les formats QCIF, SIF, et CIF respectivement. La définition standard (SD pour standard definition) est beaucoup employée pour la transmission d'images numériques issues de la télévision à tube. Les formats de haute définition (ou HD pour high definition) permettent d'afficher les images avec une grande précision. On y trouve le format 720p qui est un format intermédiaire faisant partie intégrante de la HD ; c'est un format de résolution de 1280×720 pixels en mode progressif, et est utilisé pour l'affichage des vidéos pour des applications requérantes une bande passante moins importante. Full HD est le format proposé pour tous les téléviseurs actuels avec une résolution de 1920×1080. La décennie actuelle permet l'émergence aussi d'autres formats commercialisés (voir figure II.11) comme Quad HD, Ultra HD, 4 K, et 8 K qui vont dominer le marché de la future technologie visuelle.

Format	Resolution	Nombre d'image par seconde (fps)	Mode d'affichage
Quarter CIF (QCIF)	176×144	30	progressif
Source Input Format (SIF)	352×240 352×288	30 25	progressif progressif
Common intermediate format (CIF)	352×288	30	progressif
ITU Rec. BT.601 (Standard definition)	720×480 720×576	30 25	entrelacé entrelacé
ITU Rec. BT.709 (High definition)	1280×720 1920×1080 1920×1080	24,25,30,50,60 25,30 24,25,30	progressif entrelacé progressif

Tableau II.1 Les différents formats de la vidéo numérique.

II.8 Compression et codage de vidéo

Un fichier vidéo brut exige une grande bande passante pour son transmission sur les réseaux. De même, leur stockage exige des supports à une capacité énorme.

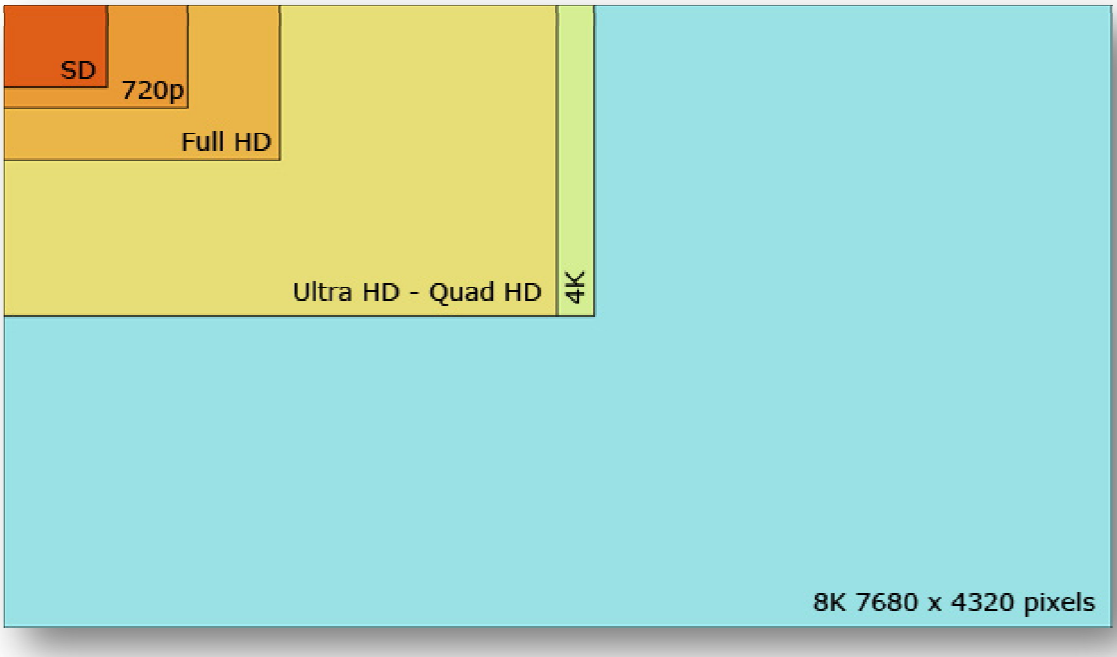


Figure II.11L'évolution des formats de la vidéo numérique : de SD au 8K.

Une seconde pour le stockage d'une simple vidéo au format QCIF sous-échantillonnée au format 4:0:0 (chaque échantillon est codé sur 8 bits) requis un espace équivalent a $(176 \times 144 \times 8 + (176 \times 144 \times 8) \times 1/2) \times 30 \approx 9123840$ bits. Une bande passante de 64 *Kbits*/*sec* ne permet pas la transmission de cette quantité, et en plus, le format brut est inapproprié pour sa transmission du a son codage compacté. Donc, la réduction de cette quantité et l'adaptation de format de flux binaire codé de la vidéo comprimée aux caractéristiques de réseaux comme la bande passante s'apparente l'unique solution qui permet de pallier cette contrainte.

La compression de vidéo exige la satisfaction de deux facteurs importants : la qualité visuelle et le débit. Une forte compression permet d'engendrer une vidéo avec une mauvaise qualité, c'est-à-dire avec une grande distorsion. Ce qui nécessite en conséquence de tenir toujours en considération le compromis débit-distorsion lors de la compression. En plus, un contrôle de débit s'avère incontournable lors de codage du flux binaire afin de faciliter son paquetage et sa transmission sur le réseau.

La dimension temporelle est la seule différence qui existe entre une image et la video. La compression de chaque image indépendamment comme dans motion-JPEG [26] n'est pas efficient, car il applique la compression JPEG pour chaque image sans

tenir en compte la corrélation inter-images. Supposant par exemple que la vidéo concerne une scène statique où aucun mouvement n'existe. On aura donc le même code de chaque image répété autant de fois. Puisque chaque vidéo permet de représenter des objets en mouvement, chaque objet qui est composé d'un ensemble connexe de pixel, subit seulement à des déplacements ponctuels avec un changement éventuel de couleurs (surtout en luminance). En conclusion, la compression d'une video doit tenir en considération la corrélation inter-trames en exploitant les mouvements d'objets afin d'éliminer la redondance temporelle, et aussi la corrélation intra-trame qui permet d'éliminer la redondance spatiale au sein de chaque image.

Le schéma commun de la compression vidéo dans tous standards existants comprend généralement les étapes suivantes (voir figure II.12) : la mise en forme, la prédiction (intra ou inter trame), la transformation, la quantification, et le codage entropique.

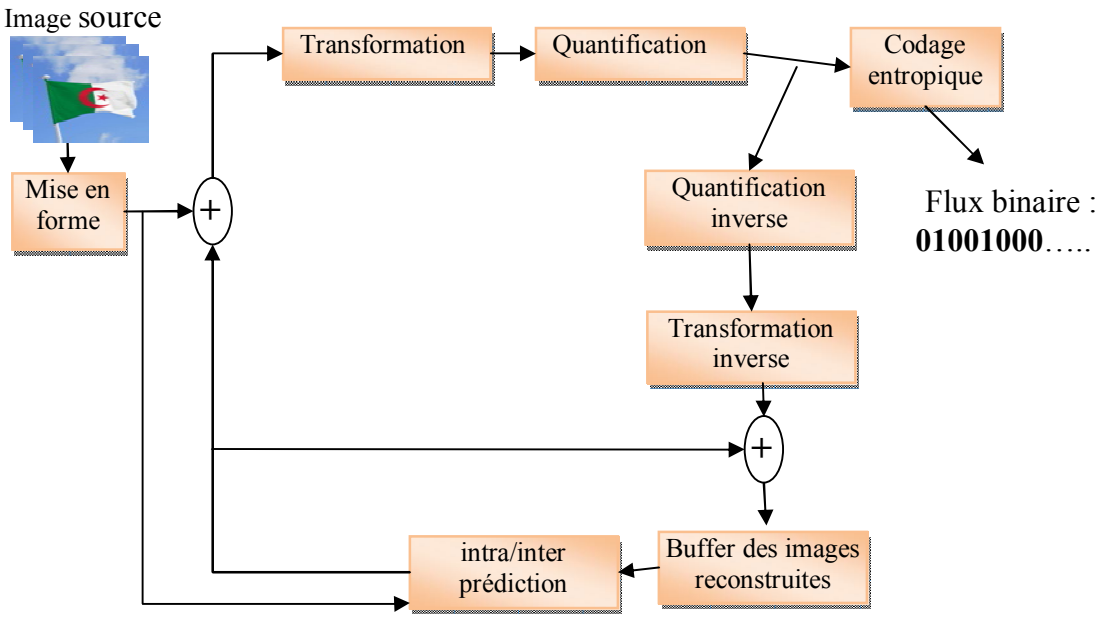


Figure II.12 Diagramme de blocs décrivant la compression vidéo.

II.8.1 La mise en forme

Chaque codeur transforme une séquence vidéo à compresser en un autre flux binaire qui respecte un format normalisé. Cette séquence est divisée en une suite de groupes d'images noté GOP dont la taille est définie auparavant comme une option configurée

pour le codeur. On distingue trois types de trames selon le type de prédiction employé :

a) Une image/trame intra prédite : notée trame I, c'est une image qui est prédite et codée a partir de ses propres données picturale sans tenir en compte les données d'autres images qui la précédent, voire qui la suivent. Chaque GOP commence toujours par une image intra, et il peut contenir une ou plusieurs trames I, et elle sert généralement pour prédire d'autres images P ou B. Son taux de compression est plus élevées par rapport a celui d'images de type P ou B.

b) Une image/trame prédite : notée trame P, C'est une image à coder qui fait l'objet d'une *prédiction avant* (forward prediction). Elle sert de référence pour prédire des images B ou des images P. La compression est nettement plus importante car on élimine seulement la redondance temporelle en exploitant la ressemblance qui existe entre les images successives.

c) Une image/trame bi-prédite : notée image B, c'est une image qui est prédite a partir d'une ou plusieurs images en combinant a la fois une prédiction avant (forward prediction) et une prédiction arrière (backward prediction). Son taux de compression est généralement moins élevé que celui de l'image P.

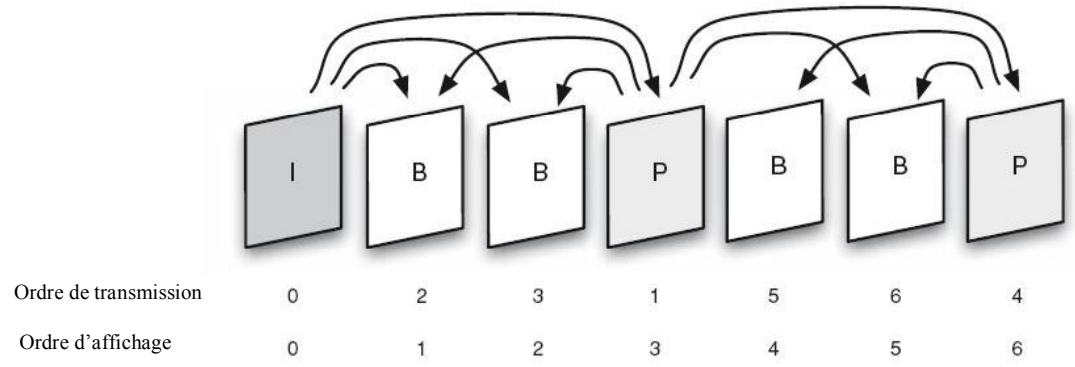


Figure II.13 L'ordre d'affichage et de transmission d'un GOP de type IBBPBP.

La figure II.13 montre un exemple d'un GOP qui est le plus répandu dans la majorité des standards de compression composé de sept images. L'ordre d'images à décoder et à afficher est $I_0B_1B_2P_3B_4B_5P_6$. Cependant, le codeur

transmet I_0 au premier lieu au décodeur, après, il transmet les données relatives à $P_3B_1B_2P_6B_4B_5$ respectivement.

Chaque image est subdivisée en tranche (slice) composée de régions rectangulaires de taille fixe. En des normes comme MPEG1, MPEG2, ou MPEG4, ces régions sont appelées macroblocs, et elles prennent une taille fixes de 16×16 pixels, alors dans d'autres normes comme HEVC [27], ces régions sont définies comme étant CTU (coding tree unit) avec une taille fixe qui varie entre 4×4 et 64×64 pixels selon la configuration choisie pour le codage. Après, le codeur décompose chaque région en structure arborescente binaire dont chaque nœud est un bloc ayant une taille qui varie entre 4×4 pixels et la taille de la région elle-même selon la politique du standard visée. Le codeur choisi parmi toutes les structures possibles celle qui permet d'avoir une meilleure qualité visuelle pour la région à décoder et un débit minimum pour son codage. Ceci est achevé en minimisant une contrainte débit-distorsion. Le codeur traite chaque région selon un parcours préfini. Différents parcours existent dont le plus connu est le parcours qui balaye l'image région par région en commençant de gauche à droite et de haut en bas.

La figure ci-dessous illustre une décomposition d'une image en slices et en macroblocs.

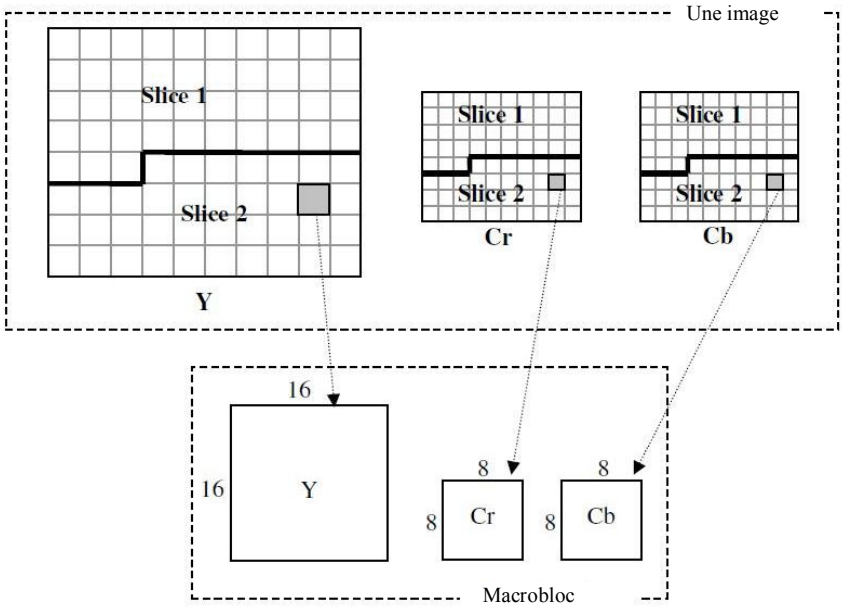


Figure II.14 La décomposition d'une image en slices et en macroblocs.

II.8.2 Mesure de distorsion

L'évaluation de la distorsion qui existe entre une image codée $\hat{I}$ et une image source à coder I est une tache très sensible, et elle dépend généralement de domaine d'application de l'image $\hat{I}$. Une distorsion peut engendrer une perte d'informations si elle est grande. Une telle perte n'est pas permise pour des images médicales, militaires, ou satellitaires. En plus, une grande distorsion qualifie tout simplement une mauvaise qualité obtenue pour l'image $\hat{I}$.

L'évaluation de la distorsion peut se faire au moyen de calculs mathématiques dans le cas d'une évaluation objective. Cependant, comme ce calcul ne reflète pas toujours la qualité d'image souhaité, l'intervention de la présence d'observateurs experts dans le domaine d'application s'avère nécessaire surtout dans le cas médical. En conséquence, cette évaluation dépend de deux types de critères : objectif et subjectif.

II.8.2.1 Les critères objectifs

On peut définir plusieurs critères quantitatifs en mesurant l'erreur entre l'image reconstruite $\hat{I}$.et l'image originale $\hat{I}$, toutes deux de dimension $M \times N$.

La valeur absolue des différences SAD (Sum of Absolute Differences) et la somme des carrés des différences SSD (Sum of Squared Differences) qui sont décrites dans les équations 2.4 et 2.5, sont deux critères qui sont majoritairement utilisés pour la recherche de meilleur estimateur prédit en intra/inter prédiction.

$$SAD = \sum_{x=1}^{M} \sum_{y=1}^{N} \left| I(x,y) - \hat{I}(x,y) \right| \quad \text{(II.4)}$$

$$SAD = \sum_{x=1}^{M} \sum_{y=1}^{N} \left(I(x,y) - \hat{I}(x,y) \right)^2 \quad \text{(II.5)}$$

Le critère historiquement utilisé pour évaluer deux images entre elles, est le PSNR pour *Peak Signal to Noise Ratio* dont l'unité est le décibel dB. Cette mesure de distorsion est donnée par la relation suivante :

$$PSNR = 10 log_{10}\left(\frac{255^2}{EQM}\right) \quad \text{(II.6)}$$

Avec EQM représente l'erreur quadratique moyenne entre Iet $\hat{I}$, et il est donné par la formule suivante :

$$EQM = \frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}\left(I(x,y) - \hat{I}(x,y)\right)^2 \quad \text{(II.7)}$$

Si l'image $\hat{I}$ est identique a I, alors la valeur de PSNR est l'infini. Une image de haute qualité signifie que son PSNR est grand. Expérimentalement, si la valeur de PSNR est supérieure à 40 dB, alors on estime que cette image a une très bonne qualité.

II.8.2.2 Les critères subjectives

L'autre démarche consiste à mettre en œuvre des essais subjectifs présentés à un nombre significatif d'observateurs ou experts dans le domaine et effectués selon une méthode aussi précise que possible. On procède ensuite à une étude statistique pour calculer les moyennes, les variances, etc. La mise en œuvre d'un test est délicate car les causes d'ambiguïté et d'incertitude sont nombreuses.

Parmi les nouvelles notions introduites pour mesurer subjectivement la qualité d'image, on trouve :

1) Le défaut juste perceptible (Just Noticeable Defect JND) [28] : la compression est optimale quand on atteint le JND pour une distance d'observation donnée.
2) la dégradation élégante (graceful degradation) : elle permet de refléter la manière dont l'observateur perçoit les défauts observés.

II.8.3 La prédiction

Afin de démineur la quantité d'informations à transmettre, la prédiction cherche à éliminer la redondance spatiale au sein de la même image dans le cas d'une intra prédiction, ou a éliminer la redondance temporelle en exploitant la corrélation qui existe entre les images successives de la séquence vidéo dans le cas de l'inter prédiction. Pour cela, on code seulement l'erreur résiduelle $e = x - \hat{x}$ qui existe entre l'échantillon source x et l'échantillon prédite $\hat{x}$ selon un critére objectif. Le décodeur reconstruit l'échantillon original x en calculant tout simplement$x = e + \hat{x}$.

II.8.3.1 L'intra prédiction

La prédiction intra exploite la redondance spatiale a l'intérieure de la même image afin de modéliser un bloc courant a partir des blocs en voisinage. Ce processus est appliqué aux données de type intra comme les images I ou les slices intra. Cependant, la première image de la séquence est obligatoirement codée comme étant image I.

Le processus d'intra prédiction varie selon la norme de compression choisi. Quant au MPEG1, le processus d'intra prédiction ressemble beaucoup à celui de la compression JPEG. Cependant, l'intra prédiction en H.264 est beaucoup améliorée par rapport a ses prédécesseurs.

H.264 utilise deux types d'intra prédiction pour prédire un macrobloc de luminance selon sa texture, et un troisième pour les chrominances. Le premier est appliqué aux macroblocs de taille 16×16 échantillons, alors le seconde est appliqué aux blocs de 4×4 échantillons, et le troisième type est appliqué aux blocs de 8×8 échantillons de chrominance. La prédiction Intra 16×16 fait recourt à 4 modes obéissant chacun à une direction et à des équations caractéristiques. La prédiction intra 4×4 est utilisée généralement pour modéliser les images texturées et elle offre neuf modes de prédictions dont chacun favorise une direction et utilise les positions de pixels voisins qui lui sont appropriés (voir figure II.15).

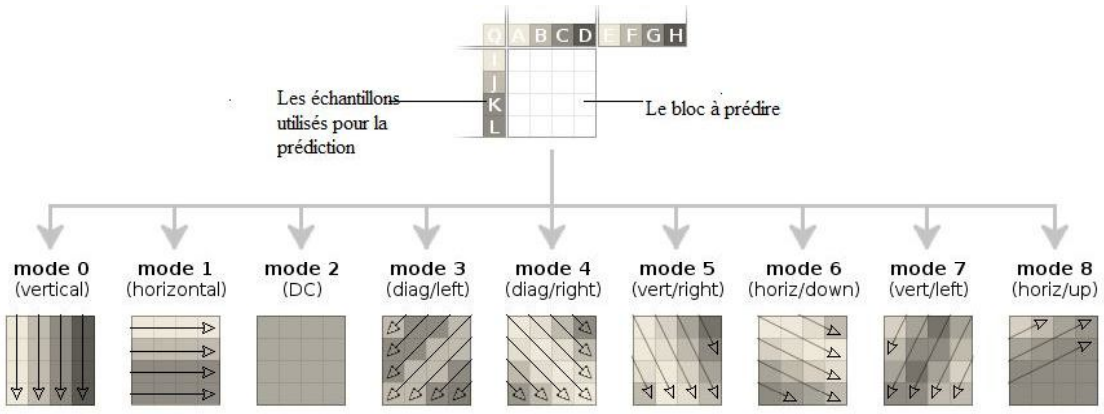


Figure II.15 Les neuf modes d'intra prédiction utilisés en H.264 pour prédire les blocs 4×4 pixels.

Pour chaque type de luminance, et en fonction de la ressemblance entre le macrobloc original et le macrobloc prédit, le mode de prédiction fournissant l'erreur résiduelle la plus faible est sélectionné. La comparaison entre deux macroblocs est calculée à

l'aide de SAD ou SAE. SAD est majoritairement utilisé pour son faible coût calculatoire. Finalement le codeur sélectionne le meilleur entre les deux types de prédiction qui offre la distorsion minimale pour le macrobloc à prédire.

II.8.3.2 L'inter prédiction

La prédiction inter cherche à éliminer la redondance temporelle en exploitant la corrélation qui existe entre les images successives dans une séquence vidéo. Cette corrélation est déterminée par l'analyse de mouvement perçu lors de défilement d'images de la séquence. Ce type de prédiction est appliquée pour les images de type P ou B a partir d'images de référence.

En général, la détection/estimation de mouvement de chaque pixel en utilisant les techniques de détection de flots optiques est très couteuse en calcul. Les techniques basées sur l'analyse de mouvement de blocs inter images permettent de réaliser le même objectif avec un cout minimum. Cette analyse dépend de l'estimation de mouvement d'un bloc courant en cherchant le meilleur bloc dans les images de référence passées et/ou futures qui lui ressemble selon un critère objectif. Et aussi, elle dépend de la compensation de mouvement qui permet de reconstruire le bloc courant à partir d'un bloc de référence.

Pour la plupart des normes de MPEG et ITU-T, la détection de mouvement a lieu sur l'entité de macrobloc. Elle consiste, connaissant un macrobloc de luminance courant, à trouver le macrobloc de luminance qui lui ressemble le plus dans une image de référence. C'est le macrobloc de référence. Connaissant la position des deux macroblocs, on en déduit un vecteur de déplacement MV (motion vector). Les critères de ressemblance entre deux macroblocs sont généralement l'erreur quadratique moyenne et SAD.

La recherche de macrobloc de référence se fait à l'intérieur d'une fenêtre dont les dimensions sont en fonction des valeurs de deux paramètres, f_x et f_x (voir figure II.16). Cette recherche peut se faire en mode image ou en mode trame. En mode image, on a un macrobloc courant de dimension 16×16 et on recherche le meilleur macrobloc dans l'image de référence. On obtient un vecteur MV ayant deux coordonnées x et y. En mode trame, on a deux macroblocs courants de dimension 16×8 correspondant à chaque trame et on recherche séparément le meilleur macrobloc

16×8 dans chaque trame de l'image de référence. On calcule donc un vecteur pour chaque trame.

Les techniques basées sur l'estimation de mouvement sont très riches. Dans l'algorithme FULLSEARCH (recherche intégrale), on cherche systématiquement le meilleur macrobloc sur toute la fenêtre, et elle donne des meilleurs résultats. Malheureusement, cette technique est très couteuse en couts de calcul.

La compensation de mouvement est la phase qui consiste à prendre le macrobloc de référence et à le déplacer selon la valeur du MV correspondant. Par exemple, au décodage, pour obtenir le macrobloc décodé, on compense le macrobloc de référence puis on lui ajoute l'erreur de prédiction.

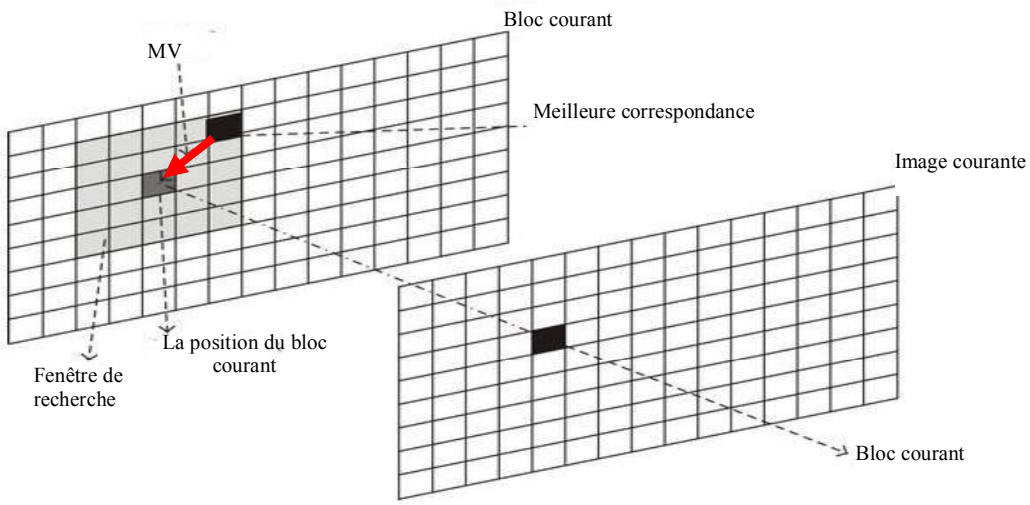


Figure II.16 Exemple d'estimation de mouvement.

Afin d'améliorer la précision de recherche de bloc de référence, l'estimation de mouvement peut se faire au demi-pixel, ou au quart-pixel, ceci permet d'obtenir un macrobloc de référence avec une précision élevée. Un exemple de l'estimation de mouvement au demi-pixel est illustré dans la figure II.17. Les positions des demi-pixels sont déduites par une simple interpolation bidimensionnelle.

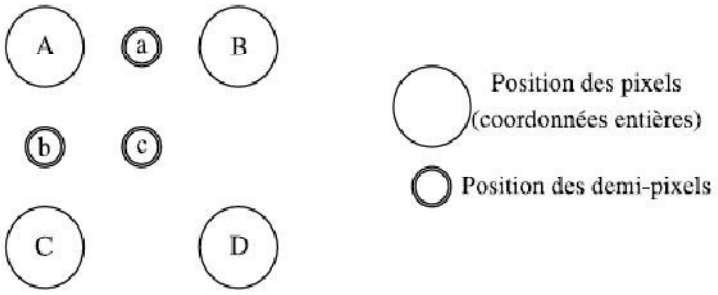


Figure II.17 estimation de mouvement a demi-pixel.

II.8.4 La transformation

Afin de réduire la redondance spatiale par éliminer au maximum la corrélation qui existe entre les pixels voisins de chaque macrobloc, l'erreur résiduelle e résultant de la prédiction (inter ou intra) est calculée pour chaque bloc constituant le macrobloc. Apres, elle est soumise à une analyse fréquentielle pour dissocier les basses fréquences des hautes fréquences.

La transformée en cosinus discrètes DCT est la plus employée dans les normes de compression vidéo comme MPEG1, MPEG2, et MPEG4. Cependant, une transformée novatrice de DCT est utilisée en H.264, elle s'agit de la transformée entière donnée par l'équation 2.8. De plus, son implémentation ne comporte que des additions et des décalages ce qui donne l'avantage de stocker des résiduels entiers et non plus flottants comme dans les prédécesseurs de H.264. Notons que cette transformée conserve les mêmes propriétés qu'une DCT classique.

$$Y = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix} \begin{pmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{pmatrix} \quad \text{(II.8)}$$

II.8.5 La quantification visuelle

Apres la transformation, l'erreur résiduelle transformée est quantifiée afin d'éliminer les hautes fréquences d'une part, et d'autre part, pour convertir l'espace des valeurs réelles prises par les coefficients en un espace de valeurs entières utilisable pour le codage entropique.

En général, les normes vidéo récentes emploient un type particulier de quantification scalaire par l'utilisation des formes matricielles de quantificateurs dont chacune est

liée à un pas de quantification QP (Quantizer step). Cette dernière détermine la qualité d'image décodée et le débit requis pour son codage. Un grand pas de quantification signifie qu'une forte quantification aura lieu, c'est-à-dire l'élimination de plusieurs coefficients, ce qui conduit a une image de mauvaise qualité après son reconstruction lors de décodage. A contrario, une image de bonne qualité implique l'utilisation d'un petit pas de quantification.

Les coefficients quantifiés QTC seront balayés ensuite selon un parcours en zigzag qui peut prendre différentes formes (pattern). Le plus répandu est celui qui commence de QTCs de basse fréquence et qui termine par les hautes fréquences.

II.8.6 Le codage entropique

La dernière étape est le codage de données compressées en flux binaire (en anglais bitstream ou binary stream) utilisable pour la transmission. Chaque norme est définie par un format standardisé de flux binaire ouvert à tous les constructeurs et à tous les développeurs avec la manière de son décodage correspondant.

Les données à encoder issues des étapes antérieures de compression (prédiction, quantification,…), et d'autres données informatives de codage/décodage comme QP, header, et EOF (end of file), seront arrangées dans des entités entropiques qui s'appellent *des éléments syntaxiques*.

Le module de codage entropique a pour vocation de transformer ces éléments syntaxiques en codes binaires qui seront empilés ultérieurement dans le flux binaire final de l'information encodée.

Le codage entropique de QTCs varie d'une norme à une autre. Des normes comme MPEG utilisent des codes à longueur variable VLC (variable length codes) sélectionnés a partir de tables de Huffman. H.264 introduit le concept d'adapter le codage d'un bloc aux statistiques de QTCs et la relation fréquentielle qui lie le bloc a coder avec les blocs déjà codé (le contexte), et ceci en utilisant un codage d'entropique de type CAVLC (Context Adaptative Variable Length coding) [29] et de type CABAC (Context-based Adaptive Binary Arithmetic Coding) [30].

CAVLC est un codage à longueur variable basé sur des tables de Huffman. Il sert à coder les QTCs non nulles selon un ordre en zigzag. Le nombre total des coefficients

non nuls Coeff_token est codé en premier lieu. Les coefficients nuls à la fin de chaque bloc 4×4 ne sont pas codés. Les zéros intermédiaires ne sont pas aussi codés. Ils sont repérés par leur position par rapport à chaque coefficient non nul Run, leurs nombre total étant la valeur Totalzeros. La figure II.18 montre le codage d'un bloc de 4×4 QTCs en utilisant CAVLC.

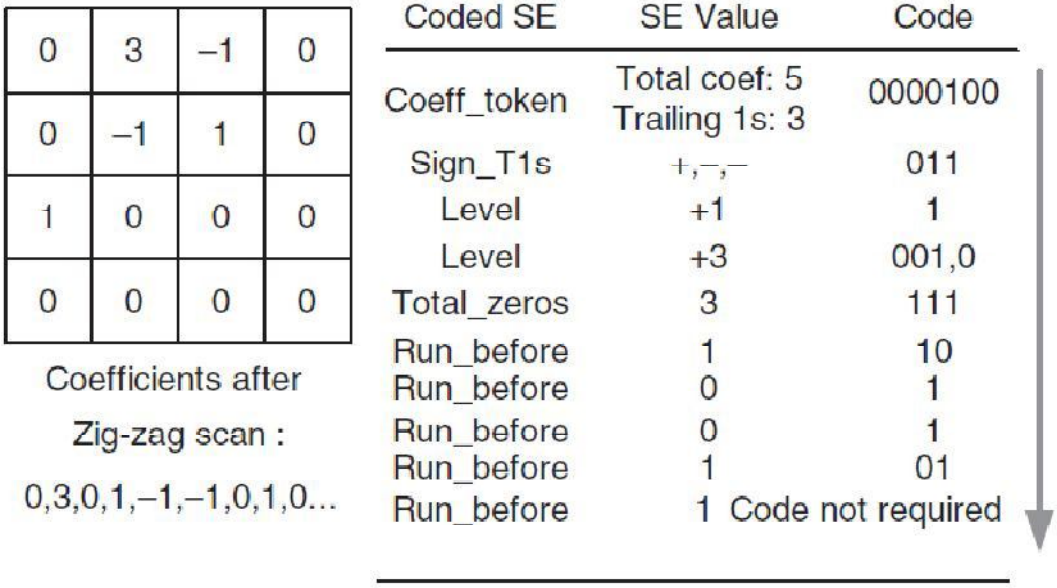


Figure II.18 Exemple de codage d'un bloc de QTCs a l'aide CAVLC [31].

CABAC est un codage entropique introduit récemment en H.264 pour améliorer la compression des différents éléments syntaxiques à coder. Il est basé sur le codage arithmétique binaire dont l'alphabet de source est un ensemble qui contient deux symboles ''0'' et ''1''. Pour cela, chaque élément syntaxique non binaire est soumis à une étape de binarisation afin de le convertir en une séquence de décisions binaires qui s'appellent des bins. Ainsi, CABAC encode les bins d'un élément syntaxique sous forme d'un nombre fractionnaire $\in [0,1[$ codé en virgule fixe sous forme d'un nombre fini et fixe de bits.

La figure II.19 résume le processus de CABAC en suivant les étapes suivantes :

1) La binarisation : elle est appliquée seulement aux éléments syntaxiques non binaires.
2) La sélection de modèles de contexte pour l'élément syntaxiques à coder : un modèle de contexte est la distribution de probabilité associée aux bins

constituant l'élément syntaxique à coder, c'est-à-dire les probabilités $p(0) et\ p(1) = 1 - p(0)$. Ces probabilité sont définies comme étant LPS (least probable symbol) pour le bin moins fréquent et MPS (most probable symbol) pour le bin le plus fréquent dans la séquence à coder. La sélection de ces modèles se fait soit à partir de tables prédéfinies de modèles de contextes pour coder les bins de certains éléments syntaxiques (regular bins), ou soit par assumer $p(0) = p(1) = 0$ au début de chaque slice. Ce dernier cas est appliqué pour l'encodage de bins d'éléments syntaxiques qui s'appellent bypass bins.

3) Le codage arithmétique binaire : le codeur arithmétique remplace chaque séquence binaire de l'élément syntaxique par un nombre fractionnaire $\in [0,1[$ codé en virgule fixe selon les modèles de probabilités choisies. Notons le codeur arithmétique devise chaque intervalle en deux sous-intervalles pour chaque bin à coder de la séquence.
4) La mise a jour de modèles de probabilité : les modèles de probabilité sont mises a jour selon les fréquences de bins récemment codés. Cette étape est intervenue seulement dans le cas de mode regulier.

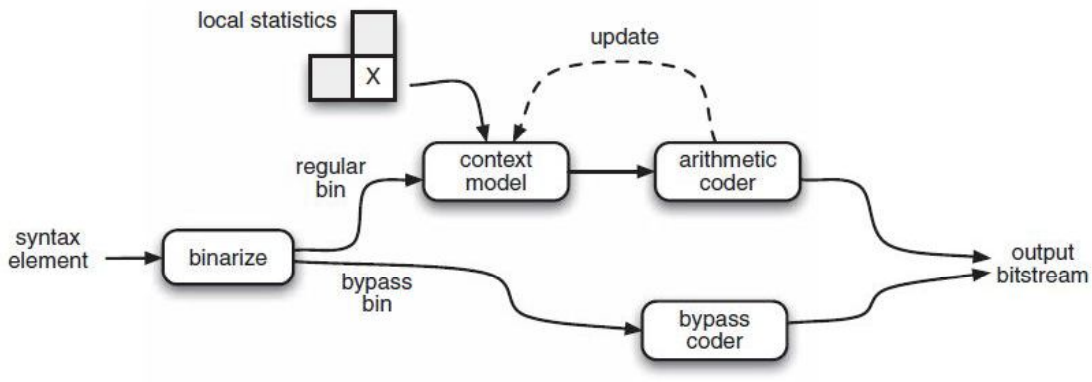


Figure II.19 La chaine de codage de CABAC [31].

II.9 Les normes populaire en codage de vidéo

La normalisation joue un rôle crucial pour le succès des standards vidéo qui sont populaire aujourd'hui. Elle permet d'offrir un choix en termes d'efficacité, interopérabilité, et fiabilité, et aussi, elle permet d'unifier les efforts et les contributions des différents chercheurs pour répondre à une large gamme de

constructeurs et d'utilisateurs. Elle commence par l'évaluation d'une norme déjà finalisée par rapport aux divers besoins des consommateurs, constructeurs, et les développements technologiques récentes afin d'initier un premier pas vers un futur standard par un brouillon de travail (working draft) ou cahier de charge a l'ensemble de chercheur d'une équipe de normalisation.

Au niveau international, les deux organismes les plus actifs dans le domaine de normalisation des systèmes de compression vidéo sont l'UIT-T et l'ISO/IEC. La figure II.20 montre les travaux antérieurs de ces deux organismes jusqu'à le lancement de HEVC. Les travaux techniques de l'ISO/IEC sont menés au sein du groupe MPEG (Motion Picture Experts Group) qui a défini les standards MPEG-1, MPEG-2 et MPEG-4 pour des applications aussi variées que la télévision ou le multimédia. En parallèle des activités de MPEG, le groupe vidéo de l'UIT-T s'intéresse principalement à la définition de recommandations techniques destinées aux applications de visiophonie et de visioconférence (normes H.261 [31] et H.263 [32]).

Ces deux organismes ont unifié leurs efforts en un seul groupe en travaillant en collaboration pour normaliser le standard H.264/AVC et la norme émergente de HEVC.

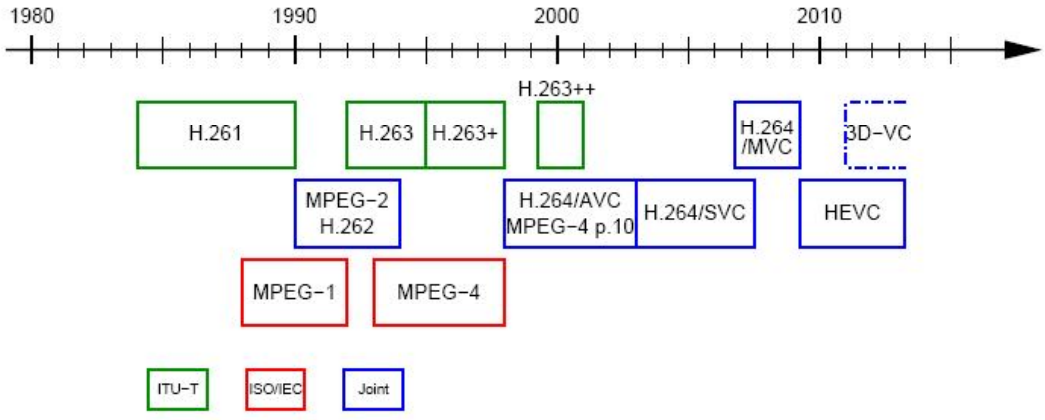


Figure II.20 les différents normes de compression video conçues par ISO et ITU-T.

II.9.1 Les normes de ITU-T

a) H.261 : vise les applications de visiophonie pour le réseau RNIS à des débits multiples de 64 kbit/s. Les formats d'image traités sont le QCIF (144x176 pixels) et le CIF (288x352 pixels). La fréquence image de base est 29.97 Hz mais peut être réduite.

b) H.263 : est une norme de codage vidéo pour la communication vidéo à très bas débit dont la première version fut adoptée en 1995. Elle vise les applications de visiophonie et de visioconférence sur RTC et RNIS. Cette norme repose sur les principes mis en place par la recommandation H.261. Les formats d'images sont des multiples et sous-multiples du CIF (352x288 pixels). La version 2 de la recommandation H.263 (1998), souvent appelée H.263+ [34], met en œuvre douze options supplémentaires et permet désormais de définir des formats et fréquences d'image personnalisés. Les caractéristiques de vidéo (Taille, fréquence) sont transmises dans le flux vidéo. Les options ajoutées améliorent fortement la qualité et la robustesse aux erreurs.

c) La dernière version de H.263 (2000), appelée H.263++ [35], ajoute trois options et une spécification à la version antérieure. Outre l'amélioration en termes de qualité et de taux de compression, elle prend mieux en compte la transmission vidéo temps réel sur des réseaux à qualité de service non garantie (IP et mobiles).

II.9.2 Les normes de ISO/IEC

a) La norme MPEG-1[17] : développé en 1998 pour répondre aux exigences liées à la sauvegarde de fichiers vidéo sur CD-ROM.

b) La norme MPEG-2 [18]: a été définie pour les applications liées à la TV numérique, à la fois au niveau professionnel (production audiovisuelle, etc.) et au niveau du grand public (diffusion vers les postes TV). Elle reprend les principes de MPEG-1 en ajoutant les outils indispensables pour les applications télévisuelles : traitement des formats entrelacés, optimisation des outils MPEG-1 (dynamique des vecteurs de mouvements, etc.), scalabilité visant la compatibilité TV/TVHD. Ce standard a été adopté par le consortium DVB (Digital Video Broadcasting) pour les services de TV numérique par

voie hertzienne terrestre (DVB-T) et satellite (DVB-S). Il est également utilisé comme format de codage du DVD (Digital Video Disc).

c) MPEG4 [19] introduit l'interaction et la création multimédia. Les audiovisuels sont structurés en objets, tout comme les langages de programmation sont devenus des *langages orientés objets*. C'est l'évolution majeure de cette version. Elle permet de gagner encore en termes de compression, mais, surtout, d'introduire l'interactivité avec l'utilisateur final à tous les instants de la chaîne : de la production à l'affichage en passant par la diffusion. Les outils de granularité, de profils et de niveaux sont étendus à tous les types d'objets. La séparation des éléments de l'audiovisuel en objets permet de mixer des objets naturels (sons naturels, photographies, vidéos, reconstructions 3D) avec des objets synthétiques (sons et images 2D/3D de synthèse).

II.9.3 H.264/AVC

Connue aussi sous MPEG-4 Part 10, Advanced Video Coding (MPEG-4 AVC). Les groupes de travail à l'UIT-T et à MPEG ont approuvé le rapprochement de leurs équipes vidéo pour la définition commune d'un nouveau standard de compression. Cette décision a conduit les groupes à fusionner sous le nom de JVT (Joint Video Team) le 6 décembre 2001. Le but de cette nouvelle entité est de standardiser un codec vidéo dont la base est H.26L. Les travaux, commencés en 1998 à l'UIT-T, devraient aboutir à un standard international en mars 2003.

La structure définie par H.264/AVC regroupe une couche vidéo (Video Coding Layer ou VCL), qui représente le contenu vidéo, et une couche réseau (Network Abstraction Layer ou NAL), qui comprend les en-têtes appropriées pour le transport de l'information par des couches transports ou des supports de stockages particulier.

La partie AVC définit trois profils :

1) le profil *Baseline* utilisé pour des applications de type vidéo-téléphonie, vidéoconférence, communication sans fil et mobile.

2) le profil *Main* utilisé pour des applications de type diffusion TV, stockage vidéo.

3) le profil *eXtended* utilisé pour des applications multimédias à haute interactivité et composées d'objets hétérogènes.

Cette norme est caractérisée par plusieurs nouveaux outils introduite dans les différentes phases de codage/décodage. En prédiction, cette norme est renforcée par plusieurs modes d'intra prédiction. La transformée entière qui est une version améliorée de la transformée en cosinus discrets est introduite pour éviter les problèmes d'arrondissement. Le codage entropique est caractérisé par l'introduction des codeurs adaptatifs aux contextes comme CAVLC et CABAC. Et aussi, l'amélioration de la visibilité lors de l'affichage de la trame décodée par l'emploi de filtre anti-bloc. Le succès de H.264 permet d'ouvrir les portes a ses extensions scalable et multi vue.

II.9.4 HEVC

HEVC est la dernière norme de compression vidéo développée conjointement par ITU-T et ISO pour faire face aux limites observées de H.264. En effet, HEVC est conçu pour être le codec de l'ère d'ultra HD.

II.10 Conclusion

En somme, la compression de l'information visuelle est achevée par combiner en cascade plusieurs outils de compression sans/avec perte. Elle commence toujours par éliminer les redondances visuelles inaperçues par le système visuel humain sans dégrader la qualité de l'information reconstruite. Ceci peut être achevé par passer par des étapes de prédiction, transformation, et de quantification. Le codage entropique quant à lui permet de convertir les données à encoder en bits qui seront empilés dans un flux binaire qui respecte un format normalisé.

Dans le cas de vidéo, nous avons vu un aperçu sur les différentes étapes de compression utilisé avec des exemples d'outils employés dans plusieurs normes de compression. La prédiction est une étape inévitable dans la compression video car elle sert d'éliminer les redondances spatiales et/ou temporelles, et de réduire la quantité transmise au décodeur sous forme d'une erreur résiduelle. Cette dernière est transformée et quantifiée afin de résulter des coefficients fréquentielles décolérés QTCs qui seront transformés en bits par le module de codage entropique.

H.264 est le fruit de travaux en collaboration entre ITU-T et ISO où elle a introduit des nouveaux outils dans tous les étapes comme la transformée entière dans l'étape de transformation, CAVLC et CABAC pour le codage entropique. En conséquence, cette norme est adoptée avec succès dans diverse applications mobiles et réseaux pour transmettre des vidéos de différentes tailles.

Dans le prochain chapitre, nous allons aborder la norme émergente HEVC qui est le successeur de H.264, où nous allons expliquer ce codec et ses caractéristiques car elle est la substance de l'évolution de toutes les normes antérieures.

Chapitre III.

Introduction a la norme émergente HEVC

III.1 Introduction

Avec l'arrivée des multimédia et de l'ère de l'information, les deux dernières décennies ont vu des développements signifiantes en normes de codage vidéo qui offrent des gains importantes en compression, et qui permettent de déployer en conséquence un grand nombre de services et d'applications numériques. La recherche sur les outils de codage de base impliqués dans la compression vidéo a commencé réellement dès les années 1950 et les années 1960 avec l'introduction de DPCM pour le codage prédictif. Pendant les années 1970, les techniques de codage basées sur la transformé et la compensation de mouvement ont été bien étudiés.

L'introduction de contenu visuel a continué à devenir une présence dominante en notre vie quotidienne, surtout avec la diversification massives des modèles d'utilisation de l'information vidéo dans des applications diverses. Les consommateurs attendent toujours plus de hautes définitions et de meilleures qualités pour leurs vidéos employées. De même, l'émission analogue des chaines télévisées avec une définition standard, et aussi les films magnétoscopes ont mené à l'émergence de la TVHD, de DVD, de Bluray, et de vidéos d'UHD qui vont dominer sans doute les futurs marchés de multimédia.

Depuis sa normalisation en 2004, la norme H.264/AVC a permis d'ouvrir les yeux aux consommateurs sur la possibilité de déployer des vidéo avec une grande résolution dans des applications réseaux et mobiles. Cependant, les vidéos aux formats hautes définition (HD, Ultra HD, 2K, 4K, 8K,…) avec une bonne qualité visuelle, exigent une très grande bande passante pour des simples utilisations sur les réseaux, et aussi des supports de capacité énorme pour leur sauvegarde.

Pour faire face a ces limites observées dans la norme H.264, et pour répondre aux exigences industriels des entreprises et/ou consommateurs, les deux premiers organismes internationaux de normalisation ISO/IEC et ITU-T ont formé une équipe commune JCT-VT (Joint Collaborative Team on Video Coding), et ont lancé en janvier 2010, un premier appel à la communauté scientifique [36] de lancement de projet HEVC (high efficiency video coding) qui va remplacer son prédécesseur H.264.

En avril 2013, la norme HEVC est standardisée après un travail énorme de l'équipe JCT-VT qui peut se résumer en plusieurs brouillons de travail WD (working draft). Elle apporte un nom formel HEVC. Cependant, le groupe ITU-T l'a standardisé sous le nom ITU-T Recommendation H.265 [37], tandis que le groupe ISO/IEC l'a nommée ISO/IEC 23008-2 (MPEG-H, Part 2). Sa standardisation va ouvrir sans doute les portes vers l'ère de l'ultra HD.

Dans ce chapitre, nous allons aborder les exigences adressés par la communauté scientifique pour le lancement de projet HEVC, un bref historique sur l'évolution de HEVC jusqu'à son standardisation, et les différentes étapes de codage associés, et les applications et les extensions qui lui sont liées.

III.2 Historique sur l'évolution de HEVC

Après la standardisation de la norme H.264/AVC en 2004, et en attendant la finalisation de leurs extensions hiérarchique SVC (scalable video coding) et multi vue MVC (multiview video coding), le groupe VCEG (Video Coding Experts Group) de ITU-T a lancé en avril 2005 un code source pour un encodeur de travail KTA (Key Technical Area) qui regroupe tous les outils possibles qui permet de tracer l'itinéraire vers la future norme de compression vidéo.

Due aux succès de la norme H.264, les consommateurs comme les entreprises, ont utilisé cette norme dans diverses applications comme la vidéoconférence, le streaming, le stockage, la conversion vers les formats haute définition pour les chaines numériques, la vidéosurveillance,…. Ces demandes massives vont dévoiler par la suite les lacunes de H.264, surtout en ce qui concerne le besoin d'un grand débit internet pour le déploiement du format HD, et aussi l'émergence des technologies perspectives qui nécessitent des formats ultra haute définition a savoir 2K, 4K, et 8K.

Les deux premiers organismes mondiaux de normalisation VCEG et MPEG ont constaté l'inévitabilité de lancer un nouveau projet du futur successeur de H.264. En janvier 2010, ils ont organisé une réunion où ils ont décidé de créer une équipe commune JCT-VC (Joint Collaborative Team on Video Coding), et ont lancé un appel a propositions CfP (Call for proposals) [36] a la commuté scientifique pour recevoir ses visions a propos de la future norme de compression et les exigences liées.

Entre 15 et 23 avril 2010, un meeting est organisé pour évaluer plus de 25 propositions sur la futur norme [38], reçues d'un grand ensemble d'acteurs scientifiques et commerciaux (chercheurs, entreprises, centres de recherches,…). Les répondeurs à CfP ont été chargés à appliquer leurs propositions sur un ensemble de 18 séquences vidéos de test (benchmark sequence), qui sont regroupées en 5 classes selon leurs résolutions allant de 416×240 jusqu'à 2560×1600 pour l'ultra haute définition (voir tableau III.1). Chacune de ces classes est accompagnée par un débit binaire probable.

Class	Rate 1	Rate 2	Rate 3	Rate 4	Rate 5
A: 2560×1600p30	2.5 Mbit/s	3.5 Mbit/s	5 Mbit/s	8 Mbit/s	14 Mbit/s
B1: 1080p24	1 Mbit/s	1.6 Mbit/s	2.5 Mbit/s	4 Mbit/s	6 Mbit/s
B2: 1080p50-60	2 Mbit/s	3 Mbit/s	4.5 Mbit/s	7 Mbit/s	10 Mbit/s
C: WVGAp30-60	384 kbit/s	512 kbit/s	768 kbit/s	1.2 Mbit/s	2 Mbit/s
D: WQVGAp30-60	256 kbit/s	384 kbit/s	512 kbit/s	850 kbit/s	1.5 Mbit/s
E: 720p60	256 kbit/s	384 kbit/s	512 kbit/s	850 kbit/s	1.5 Mbit/s

Tableau III.1 Les différentes classes de vidéo utilisées pour l'évaluation des propositions de CfP [36].

Aussi, différentes conditions de tests sont exigées pour le futur standard de compression, parmi eux on peut citer :

- la performance de compression : HEVC devrait réduire la taille de flux binaire en moitié avec une qualité visuelle équivalente a celle de High profile de H.264/AVC.
- le format de l'image : L'effort de développement de HEVC se concentrera sur un ensemble de formats rectangulaires de l'image qui incluront tous les formats utilisés généralement pour la compression, s'étendant au moins de VGA à 4Kx2K, et se prolongeant potentiellement à QVGA et à 8Kx4K.

- Des formats d'images de taille arbitraire seront également soutenus dans des limites spécifiques pour chaque niveau de compression. Le codec de HEVC soutiendra au moins la même gamme de formats d'images soutenus par la syntaxe de H.264/AVC.
- l'espace de couleurs et sous-échantillonnage de vidéos numériques a compressé. HEVC devrait supporter aux moins :
    a) l'espace de couleur YCbCr.
    b) les formats de sous-échantillonnage 4:2:0,4:2:2, ou 4:4:4.
    c) 8/10/14 bits pour le codage de chaque composant de couleur.
    d) l'encodeur peut supporter aussi l'espace RGB.
- nombre d'images par secondes (frame rate) : HEVC devrait supporter un débit qui commence de 20 jusqu'à 60 images par secondes. Cependant, ce nombre pourra atteindre jusqu'à 150 pour des applications spécifiques.
- les techniques de balayages (scanning methodes) : le balayage progressif est exigé pour tous les profils et pour tous les niveaux.
- La complexité : HEVC devrait tenir compte de la complexité logicielle et/ou matérielle pour sa faisabilité selon les contraintes de la technologie actuelles et/ou futures (temps d'exécution pour des applications temps réel, la configuration matérielle requises, bande passante, protocole réseaux surtout de transport, codage en virgule fixe ou flottante).
- HEVC devrait supporter le mode *low delay* qui est un ensemble de conditions exigeant un retard algorithmique pour le décodage de chaque image de GOP. ce mode est préféré pour les applications de communication en temps réels.
- HEVC devrait supporter aussi le mode *random access* qui permet une transition rapide lors de la sélection d'une image de flux vidéo. Ce mode est préféré pour des applications de sauvegarde et de vidéosurveillance.
- Transmission sur les réseaux : des méthodes visuelles de segmentation et de paquetage de bitstream pour les réseaux devront être développées. La couche vidéo devrait être conçue de telle sorte que les mesures appropriées de résilience d'erreur peuvent effectivement être appliquées à la couche réseau.

La sélection finales a pris en considération la qualité de la vidéo à travers les diverses classes, et chaque candidat est chargé d'implémenter ses propositions dans un logiciel de référence JM 15.

L'évaluation des propositions ont été effectués jusqu'à Mars 2010 dans trois laboratoires d'essais qui sont : FUB (Fondazione Ugo Bordoni, Rome, l'italie), EPFL (École Polytechnique Fédérale de Lausanne, Lausanne, la suisse), et EBU (European Broadcasting Union, Geneva, la suisse).

Finalement, les meilleurs propositions approuvées après l'évaluation, ont permis à l'équipe JCT-VT de décider de nommer le future projet de normalisation ''High efficiency video coding'' (ou HEVC), avec l'initiation d'un logiciel de référence TMuC[4] (Test Model under Consideration), et aussi la publication de la première version de brouillon de travail WD ver1 [39] qui explique les différents outils de décodage acceptés par JCT-VT, et le format de flux binaire décidé.

Après la publication de WD ver1, plus de 15 brouillons de travail ont été achevés afin de dresser les traits caractéristiques de HEVC. L'optimisation de la contrainte calculatoire débit-distorsion occupait un espace de travail considérable. Aussi, le codage entropique est passé par beaucoup d'améliorations.

En Janvier 2013, le brouillon final FDIS (Final Draft International Standard) de HEVC était prêt pour la standardisation. Le premier brouillon standard DS (Draft standard) [37] est publié en Avril 2013 ; il contient le format de flux binaire, la syntaxe et la sémantique des éléments syntaxiques à décoder, et l'algorithme proposé pour son décodage. Le codage est laissé aux constructeurs pour concrétiser leurs implémentations, à condition qu'elles soient conformes au format de standard de HEVC.

III.3 Les étapes de codage/décodage de HEVC

Comme chaque norme de codage vidéo, HEVC [27] dispose de son schéma de compression et aussi, de son vocabulaire. Son schéma est illustré dans la figure III.1 et il comporte les étapes ordinaires de codage vidéo y compris le découpage en unité de codage, la prédiction (intra ou inter), la transformation, la quantification, et le codage entropique. HEVC ajoute quant a lui des outils nouveaux et/ou améliorés de l'existant comme le filtre de contrôle/déblocage afin d'améliorer la qualité de l'image reconstruite, et CABAC qui est le seul codeur entropique utilisé par HEVC,…

[4] http://www.h265.net/2010/06/introduction-to-tmuc.html

Chaque image est soumise à une étape de prédiction (inter ou intra) où la première image de la séquence est toujours codée en mode intra. Dans le cas où l'image est de type P ou B. Elle est prédite à partir d'images de référence reconstruites et sauvegardées dans un entrepôt d'images de référence DPB (Decoded picture buffer). L'inter prédiction n'est que l'estimation et la compensation de mouvement. L'erreur résiduelle générée par la prédiction est transformée et quantifiée en même temps par l'emploi d'une alternative améliorée de la transformée entière, qui est le RQT (Residual quad-tree transform). Après, une étape de codage entropique est intervenue pour coder les données de la vidéo compressée (données de prédiction, coefficients issues de RQT, …) en un flux binaire décodable selon la norme HEVC.

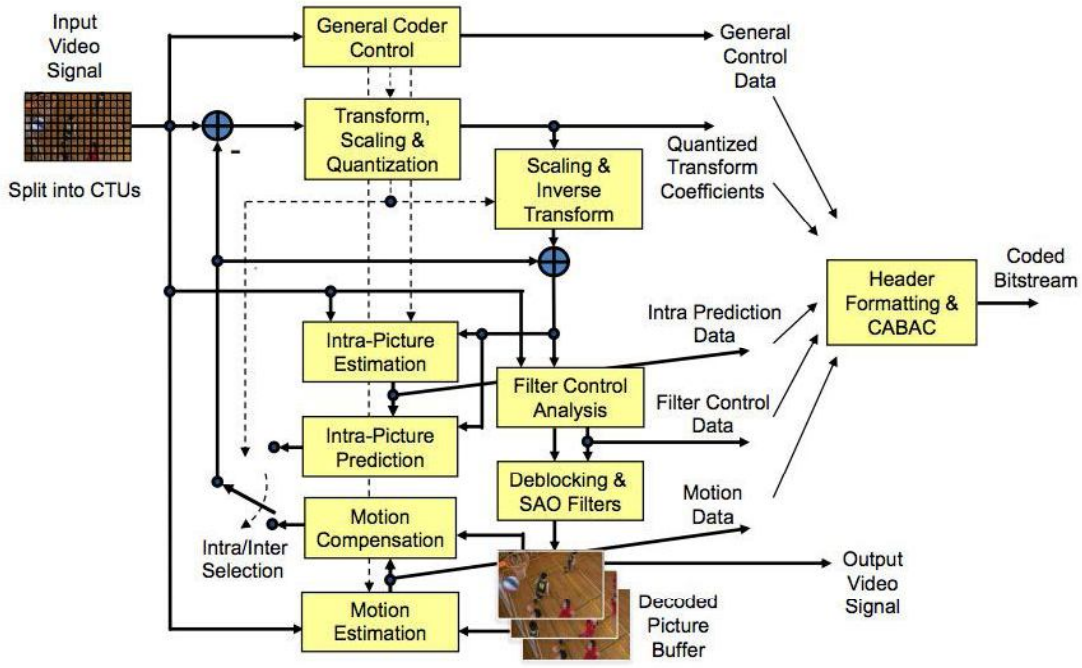


Figure III.1 Le schéma général de codage d'une séquence vidéo en utilisant HEVC.

Chaque image est découpée en régions rectangulaires qui s'appelle CTUs (coding tree unit) dont la taille est fixée et configurée pour le décodeur (voir figure III.2) ; elles varient de 4×4 jusqu'à 64×64. CTU représente l'unité de base de codage pour HEVC. Pour faciliter sa transmission au décodeur, HEVC segmente l'image en tranche (slice) ou en tuile (tile) (illustré dans images a et b de la figure III.3). Afin de faciliter son décodage, HEVC peut parcourir les lignes de CTUs selon un mode de

parcours parallélisable qui s'appelle wavefront ; les lignes de CTUs sont traités séparément comme est illustré dans la figure 3.3 (c).

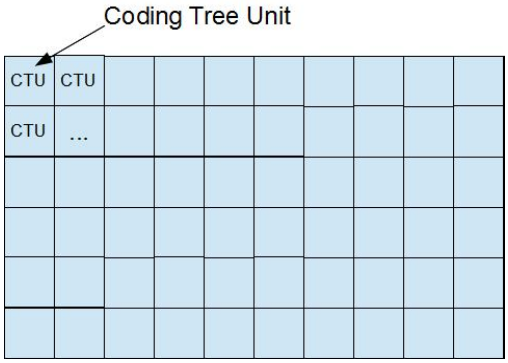


Figure III.2 Coding tree unit.

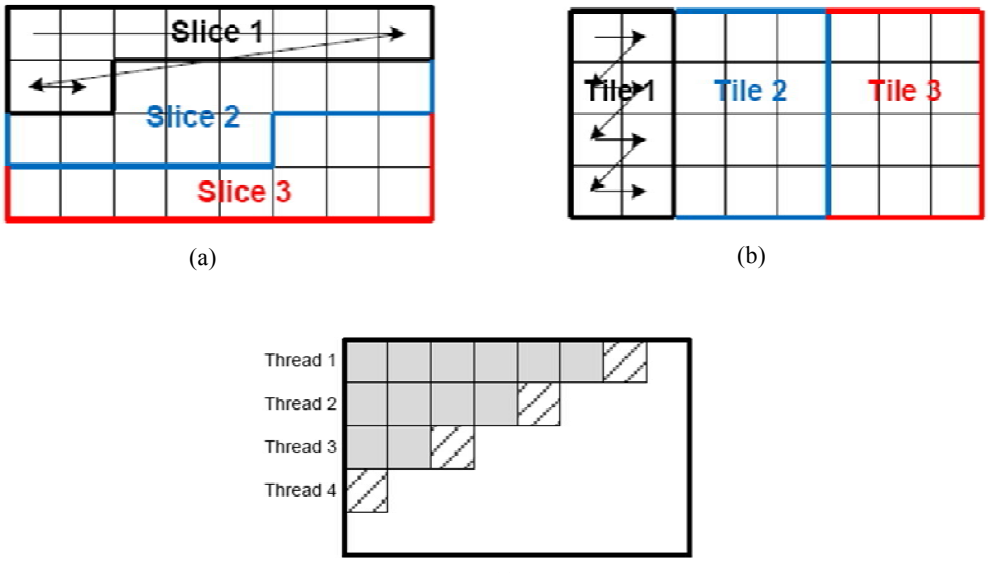


Figure III.3 La segmentation d'une image en : (a) Slice, (b) Tile, et (c) décodage en wavefront.

De même, chaque CTU a ses blocs pour chaque composante de couleur qui sont notée CTBs (coding tree bloc). Si le format de sous-échantillonnage 4 :2 :0 est sélectionné. Un CTU a un CTB de luminance de taille $N \times N$, et deux CTBs de chrominance de taille $N/2 \times N/2$.

L'optimisation de la contrainte débit-distorsion est achevé au moyen de l'introduction de trois nouveaux éléments dans le vocabulaire de codage : unité de codage CU (coding unit), unité de prédiction (PU), et unité de transformée (TU).

Chaque CTU est partitionné récursivement en petites unités qui s'appellent CUs dont sa taille varient de 4×4 jusqu'à la taille de CTU (figure III.4). Ce partitionnement se fait au moyen d'un arbre binaire dont les feuilles optimisent la contrainte débit-distorsion.

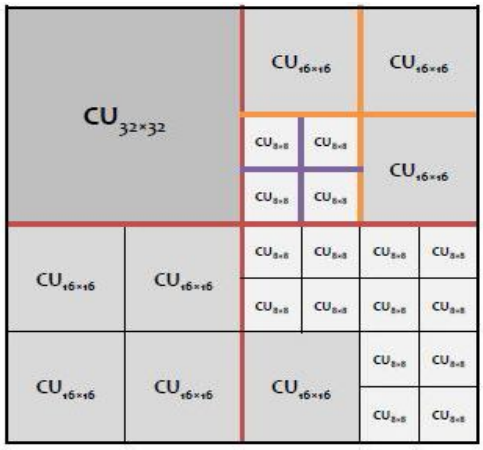


Figure III.4 Un CTU partitionné en CUs.

Il existe plusieurs structures de partitionnement en CUs que peut porter CTU. HEVC sélectionne la structure qui minimise la quantité $\sum_{i=1}^{n} D_i + \lambda R_i$, n représente le nombre de CUs composants la structure, D, R, λ représentent respectivement la distorsion (Distortion), le debit (Rate), et un coefficient de pondération λ qui s'appelle le multiplicateur de Lagrange. Cette optimisation permet de segmenter une image en blocs de texture homogène (figure III.5).

La lecture de cette structure se fait selon un parcours qui s'appelle Z-scan. Il balaye les CUs de gauche à droite et de haut en bas dont chacune sera libellée par une étiquette alphabétique comme le montre l'exemple de la figure III.6. Dans cet exemple, Les CUs sont codées et décodées selon un parcours qui commence de CU libellée (a) et se termine par la CU libellée (v)

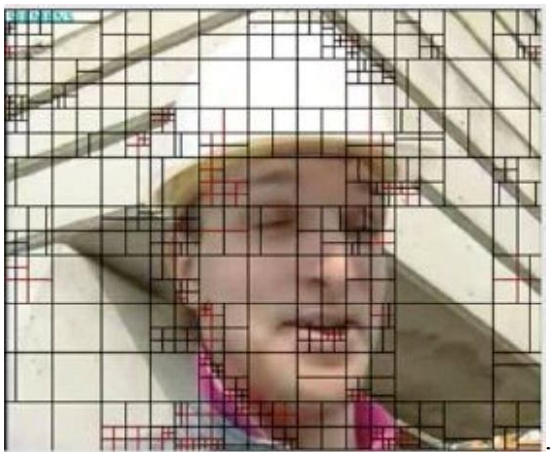


Figure III.5 Partitionnement d'une image en CUs par le codeur HEVC.

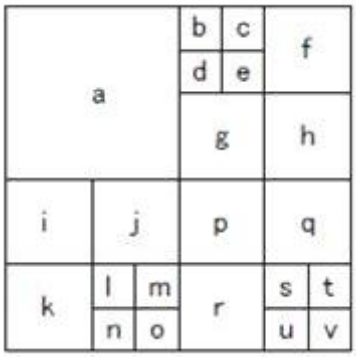


Figure III.6 Exemple d'un étiquetage alphabétique de CUs qui composent une CTU.

L'optimisation est assurée aussi par l'introduction de deux autres unités indépendantes qui sont l'unité de prédiction (PU) et l'unité de transformée (TU). La première sert à minimiser la distorsion visuelle, tandis que la deuxième sert à réduire le débit de QTCs codés au minimum.

Les PUs sont des unités qui contiennent des données relatives a la prédiction comme le mode de prédiction choisi pour la prédiction intra, les vecteurs de mouvement,….Les PUs portent une des formes rectangulaires de taille N×N ou N/2×N/2 si la prédiction est intra, ou une de huit formes de partitions rectangulaire représentés dans la figure III.7 si la prédiction est inter.

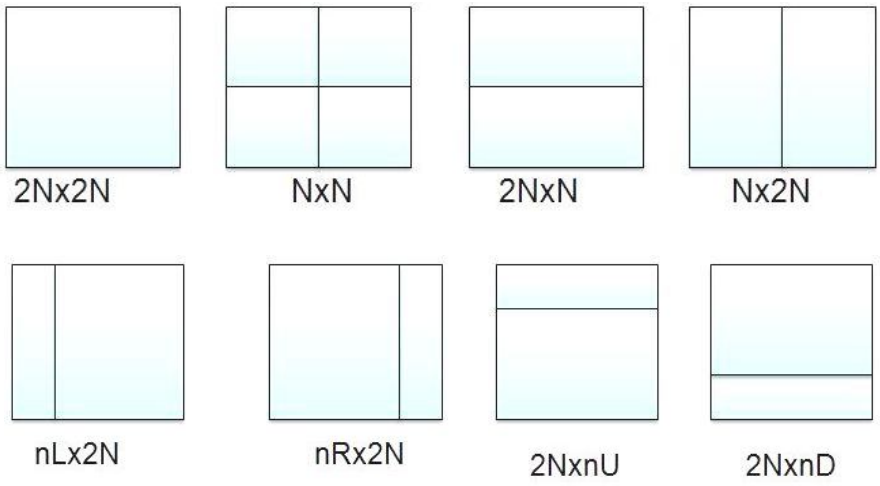


Figure III.7 Les formes utilisées que prennent les PUs en inter prédiction.

Chaque PU est partitionnée au moyen de la transformée fréquentielle RQT [40] en structure arborescente dont les feuilles sont des unités de transformée TUs. Chaque TU contient les données fréquentielles de l'erreur résiduelles relative pour chaque composante.

III.4 La prédiction

La première image de chaque séquence vidéo doit être codée en mode Intra, car elle sert comme une image de référence pour prédire les futures images de la séquence. Le but de la prédiction intra est d'éliminer la redondance au sein de l'image elle-même, et de coder les blocs de l'image en fonction de blocs se trouvant en voisinage.

Pour une composante couleur donnée, l'encodeur de HEVC partitionne chaque CB en blocs de prédiction PB (prediction block) dont la taille de chaque bloc est soit N×N ou N/2×N/2, Et il choisi la structure qui donne la somme d'erreurs $\sum_{i=1}^{n} SAD_i$ minimale, avec n représente le nombre de blocs composant la structure en blocs, SAD est La valeur absolue des différences calculée par l'équation 2.2 de chapitre III.

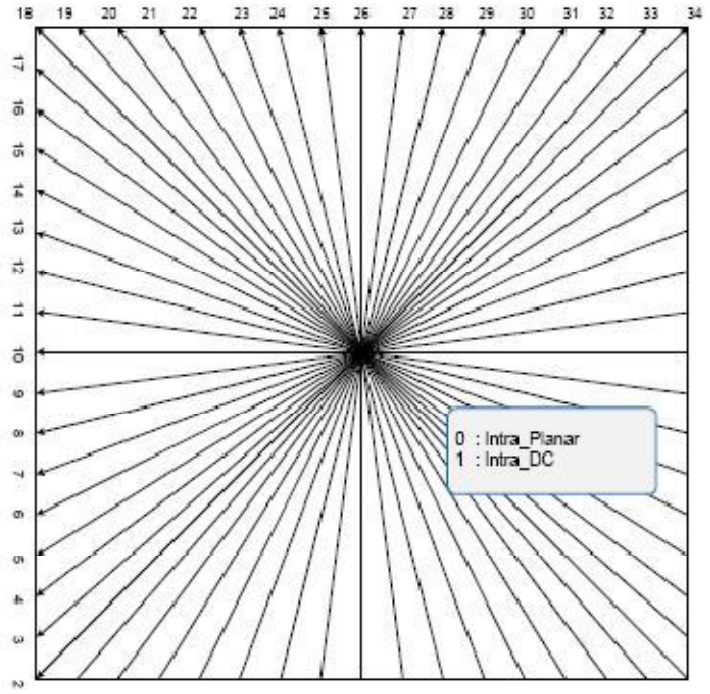


Figure III.8 Les modes directionnels utilisés par HEVC pour la prédiction intra.

L'encodeur HEVC utilise un ensemble de 35 modes de prédiction (voir la figure III.8). 33 de ce modes sont des extrapolations directionnelles a des angles variées, et les deux autres modes sont appelés le mode Intra_Planar et Intra_DC respectivement.

Pour un PB donnée, l'encodeur HEVC sélectionne le mode intra qui donne l'erreur SAD minimale. L'ensemble de modes à tester varie selon la taille de PB comme le montre le tableau III.2.

La taille de PU	Nombre de modes Intra utilisés
4	17
8	35
16	35
32	35
64	3

Tableau III.2 Le nombre de modes d'intra prédiction requis selon la taille de PU.

La prédiction inter quant a elle a connu des améliorations profondes et spécifiques. A partir d'un bloc donné, l'encodeur recherche un bloc le plus ressemblant à partir d'images reconstruite passées et/ou future dans une fenêtre de recherche dont le centre est la position de bloc courant. HEVC utilise deux types d'image pour la prédiction

inter : les images de type P qui sont souvent utilisées en mode de low delay, et les images de type B qui sont souvent utilisées en mode random access. Contrairement aux normes de codage antérieures, les images de type B sont codées à partir de deux ou plusieurs images de référence. Ces images de référence sont des images reconstruites à partir des images déjà codé, et elles sont sauvegardées dans deux listes L_1 et L_2 qui sont entreposées dans un fichier temporaire DPB. L'encodeur HEVC partitionne chaque CB en blocs PBs dont les partitions sont montrées dans la figure III.7.

L'estimation de mouvement est affinée par l'ajout de la représentation à une précision de ¼ pixels et la représentation au $1/8$ pixels de précision.

III.5 La transformation/quantification

L'erreur résiduelle d'un bloc PB issue de l'étape de prédiction est partitionnée quant a lui, en blocs de taille qui varient de 4×4 jusqu'à 32×32. Ce partitionnement est effectué au moyen d'une transformée basée sur une représentation arborescente qui s'appelle RQT (residuel quadtree transform) [40]. Les feuilles de RQT sont appelée TB (transform block), et elles sont passée au domaine fréquentielle par l'application d'une transformée variante de DCT déjà appliquée en H.264, qui est sans doute la transformée entière. L'intérêt de la transformée entière est qu'il ne nécessite pour son calcul que des opérations élémentaires comme l'addition et le décalage binaire. L'introduction de RQT et la décomposition de chaque PB en un ou plusieurs TBs a pour objectif d'adapter la transformée entière aux caractéristiques fréquentielles de l'erreur résiduelle prédite surtout le contrôle de sa résolution fréquentielle.

HEVC applique la transformée entière aux blocs de taille allant de 8×8 jusqu'à 32×32 a l'aide de l'application matricielle suivante :

$$Y = H \times X \qquad \text{(III.1)}$$

avec Y, H, et X représentent l'erreur transformée, la matrice de la transformée, et l'erreur respectivement.

Les blocs de taille 4×4 sont transformée au moyen de la transformée en sinus discrètes DST. Ceci est achevé en appliquant la matrice H suivante dans l'équation III.2 :

$$H = \begin{pmatrix} 29 & 55 & 74 & 84 \\ 74 & 74 & 0 & -74 \\ 84 & -29 & -74 & 55 \\ 55 & -84 & 74 & -29 \end{pmatrix} \quad \text{(III.2)}$$

L'erreur transformée Y est soumise souvent à une étape de mise en échelle (scaling) pour assurer que la représentation des coefficients est possible sur des architectures matérielles requérantes une représentation binaire en virgule fixe.

HEVC quantifie l'erreur transformée Y en utilisant un nouveau type de quantification scalaire contrôlé par un pas de quantification QP qui s'appelle AQMS (Adaptive Quantization Matrix Selection). Elle permet de sélectionner une matrice de quantification Q qui offre un gain supérieur en qualité visuelle. Après la sélection de la matrice Q, la quantification n'est que :

$$Z_{ij} = round(Y_{ij}/Q_{ij}) \quad \text{(III.3)}$$

Le pas de quantification varie entre 1 et 51, et plus le pas augmente, plus la qualité visuelle décroit.

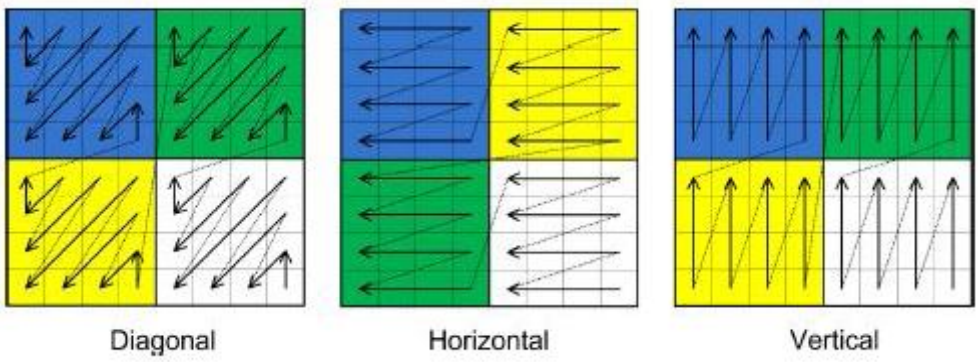


Figure III.9 Les modes de parcours possibles pour le balayage de QTCs d'un TB de taille 8×8.

Les TBs de taille supérieurs sont devisées en blocs de taille 4×4. HEVC parcourt les QTCs de chaque TB en commençant par le dernier QTC non nul de bloc inferieur gauche, et il se termine par le DC de premier bloc, et ceci au moyen de trois formes de balayage en zigzag a savoir le parcours en diagonal, le parcours en horizontal, et le parcours en vertical (voir la figure III.9). Chacun d'entre eux est sélectionné selon le mode de prédiction utilisé.

III.6 Le codage entropique

Comme toutes les normes de compression, la dernière étape sert à transformer les données compressées en bits empilés ultérieurement dans un flux binaire lisible selon un format normalisé. Dans le dernier document de la norme HEVC, on trouve seulement le format de flux binaire qui commence évidemment par l'entête et se termine par EOF (end of file), et la manière conforme de son décodage. L'entête couvre toutes les informations utiles concernant le format de la vidéo à comprimer, et la configuration employée depuis le codeur. L'architecture générale de flux binaire est hiérarchique, où il apparait les éléments syntaxiques de CTU, après, on trouve, les ceux de CUs composantes la CTU, et pour chaque CU, on trouve les éléments syntaxiques de PUs et TUs respectivement.

CABAC est le seul codeur entropique utilisé dans la norme HEVC. Il garde les mêmes étapes qui sont la binarisation, le choix de table de contexte et finalement le codage arithmétique binaire. Son principal avantage est la réduction de tables de contextes comparant par rapport à H.264.

HEVC utilise cinq types de code pour la binarisation des éléments syntaxiques non binaires :

a) code *unaires U* (*unary*) : le code unaire d'un élément syntaxique x est une chaine composée de x ''1'' terminée par un ''0''.
b) codes *unaire tronqué TU* (*truncated unary*) ; est un cas particulier de codes unaires conditionné par un paramètre C_{max}. Le code unaire est utilisé pour tout $x \leq C_{max}$. Autrement, le ''0'' est négligé et le code est juste un C_{max} ''1''.
c) code *d'Exp-Golomb de $k^{éme}$ ordre (noté EG_k)* : Ce code est une concaténation de préfixe et de suffixe. Le préfixe s'obtient par la binarisation de la partie $l(x) = \log_2(\frac{x}{2^k} + 1)$ au moyen de codage unaire. Le suffixe est la représentation binaire de $x+2^k(1-2^{l(x)})$.
d) code de longueur fixes FLC : la chaine de ce code est la représentation binaire de x avec un nombre de bits égal à $l=\lceil \log_2(cMax+1) \rceil$.
e) code *de Golomb-Rice d'ordre k (noté TR_k pour truncated rice code):* la chaine de ce code est une concaténation d'un préfixe q suivie d'un suffixe p. Etant donné un paramètre$k$, le préfixe est calculé comme le code unaire de

quotient $q = round(x/2^k)$. Alors que le suffixe est juste les k bits de la partie restante $x - q \times 2^k$.

Finalement, on note que le format de flux binaire HEVC est approprié pour la transmission sur le réseau au moyen de la structure syntaxique NAL (network abstraction layer) qui adapte la couche réseau a la couche vidéo pour faciliter la transmission de flux vidéo sous forme de paquets indépendantes.

III.7 Conclusion

Nous avons abordé dans ce chapitre la norme émergente HEVC, où nous avons donné seulement un aperçu sur les principaux outils de compression utilisés.

Après la compression et le codage, il est d'usage de sécuriser le flux binaire de la vidéo comprimée à l'aide de techniques de sécurité informatique comme cryptographie. Le prochain chapitre va jeter lumière sur comment protéger le contenu informatif de vidéo comprimée a travers un état de l'art consistant.

Chapitre IV.

Sécurité et protection de l'information vidéo

IV.1 Introduction

La croissance exponentielle d'applications qui emploie les vidéos numériques en sauvegarde sur des supports numériques comme en transmission sur les réseaux, nécessite la sécurité de l'information vidéo communiquée. Pour cela, on devrait utiliser des techniques de gestion de droits numériques DRM (*digital rights management*) afin de protéger l'information transmise.

La protection contre copie et la protection commerciale par zone sont aussi parmi les défis à surmonter quant aux applications de sauvegarde. Certains constructeurs incluent des mesures de sécurité en matériel à sauvegarder (DVD, HD DVD et Blu-Ray) pour protéger le contenu de la vidéo. D'autres constructeurs accompagnent les mesures de protection matérielle par d'autres mesures logicielles comme le chiffrement. La vidéo numérique sous ses différentes normes de compression variées est omniprésente en applications mobiles et réseaux. En diffusion télévisuelle, différentes chaines numériques payantes assurent ses droits par des systèmes d'accès conditionnel CAS (Conditional Access System) intégrés sur des cartes a puce ou sur un périphérique de démodulateur associé. Les applications de diffusion en continu ou streaming occupent une large utilisation dans la majorité des sites web (par exemple youtube), dans des applications de conversation multimedia comme en skype, ou dans des applications mobiles 3G. Ces applications récemment parlées, sont protégé par des techniques de sécurité matérielle et/ou logicielles différentes.

Le chiffrement (le cryptage) de vidéo est l'un des mesures de sécurité logicielle qui est employé massivement afin d'assurer une meilleur protection. Il permet de transformer le contenu d'une vidéo claire en inintelligible pour toute personnes non habilité à lire la vidéo claire. Et ceci est achevé au moyen d'un algorithme publique et une clé secrète de taille très réduite (nombre fini de bits). Sans ce dernier moyen,

l'utilisateur ne peut pas accéder au contenu clair de la vidéo car le contenu de la vidéo chiffrée sera en conséquence inintelligible et/ou illisible. Le chiffrement peut s'appliquer pour dégrader la qualité de vidéo, changer entièrement le contenu, ou modifier carrément le format du flux.

Dans ce chapitre, nous commencerons par donner des définitions et vocabulaires relatives au domaine de la cryptographie moderne. Tout en restant général, on abordera également les algorithmes de chiffrement symétrique et asymétrique en citant quelques algorithmes classiques et populaires. Après, on passera aux techniques de chiffrement appliquées pour la protection de vidéo numérique. Ces techniques dépendent inévitablement de la norme de codage de vidéo, et elles peuvent être appliquées avant, durant, ou après la compression. Finalement, on terminera par une conclusion par l'illustration des défis à surmonter lors de la protection de la norme récemment standardisée HEVC.

IV.2 Introduction a la cryptographie : vocabulaire et définitions

Apres avoir compressé le message (texte, image ou vidéo) et apporté le format approprié, la transformation en format incompréhensible est nécessaire afin de garder le message secret pour toutes les personnes non autorisée excepté le destinataire. Ceci est possible grâce aux techniques de science de secrets ou tout simplement la cryptographie [42].

La cryptographie est l'étude des cryptosystèmes (méthodes de chiffrement/déchiffrement) donnant la possibilité d'envoyer les données de message de manière confidentielle sur un support donné. Le chiffrement consiste à transformer une donnée (texte, message, ...) afin de la rendre incompréhensible et inintelligible pour une personne autre que celui qui a créé le message et celui qui en est le destinataire. La fonction inverse permettant de restituer le message original est appelée le déchiffrement.

La cryptanalyse est l'art pour une personne non habilité dit un modèle d'adversaire, de décrypter, de décoder, de déchiffrer le message lors de sa transmission au destinataire. C'est donc un ensemble de techniques d'attaque d'un cryptosystème pour atteindre le message d'origine.

On définit aussi la cryptologie comme étant l'ensemble des techniques combinant à la fois la cryptographie et la cryptanalyse.

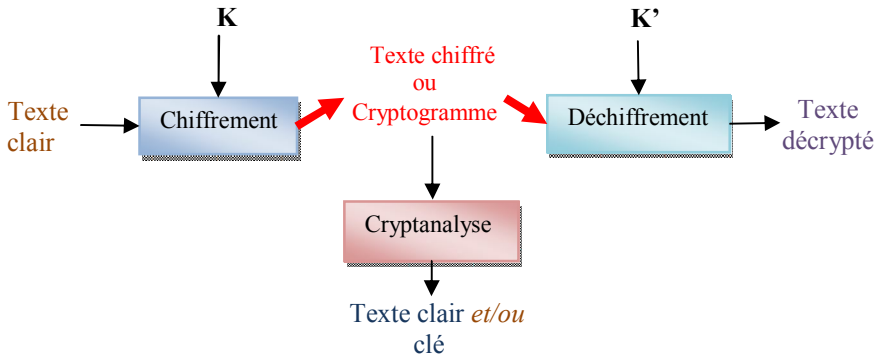


Figure IV.1 Le schéma général d'un système cryptographique.

Le but d'un cryptosystème est de chiffrer un texte clair M (le message à crypter) en un cryptogramme C (le message chiffré) au moyen d'une clé K. Les propriétés de base sont illustrées dans la figure IV.1, et ont défini mathématiquement comme suit :

$$C = E_K(M) \qquad \text{(IV.1)}$$

$$M = D_{K'}(C) \qquad \text{(IV.2)}$$

où:

- M représente le texte en clair (plaintext).
- C le cryptogramme (ciphertext).
- $K, et\ K'$ sont les clés de cryptosystème.
- $E\ et\ D$ représentent les fonctions de chiffrement et de déchiffrement respectivement.

Ainsi, le cryptosystème vérifie la propriété suivante :

$$M = D_{K'}(E_K(M)) \qquad \text{(IV.3)}$$

En outre, les fonctions de cryptosystème $E\ et\ D$ peuvent fonctionner de deux façons :

- En continu : chaque nouveau bit de flux binaire communiqué de texte clair comme en cryptogramme est traité directement.

- Par bloc : Le message est partitionné en blocs de taille fixe, et les fonctions de cryptosystème agissent directement sur ces blocs selon un mode d'opération donné.

Les qualités visées pour un cryptosystème donné dépendent de contraintes suivantes :

- Confidentialité : seules les personnes autorisées ont accès au contenu de cryptogramme.
- Intégrité des données : le message ne peut pas être falsifié sans qu'on s'en aperçoive.
- Authentification : le cryptosystème doit prendre les mesure nécessaires pour l'arrivé de message au destinataire approprié. De plus, le destinataire doit être sûr que le message reçu soit transmis depuis l'émetteur. Autrement dit, c'est la vérification et la sureté de l'identité de message.
- Non répudiation : elle signifie que ni l'émetteur, ni le récepteur ne doit nier l'émission/réception de message concerné.

En plus de ses qualités, la sécurité d'un cryptosystème possède plusieurs aspects à savoir :

- La sécurité inconditionnelle qui ne préjuge pas de la puissance de calcul du cryptanalyste (l'attaqueur) qui peut être illimitée.
- La sécurité calculatoire qui repose sur l'impossibilité de faire en un temps raisonnable, compte tenu de la puissance de calculateurs disponible, les calculs nécessaires pour décrypter un message.
- La sécurité prouvée qui réduit la sécurité d'un cryptosystème à un problème bien connu réputé difficile.
- La confidentialité parfaite où aucune combinaison de (M, C, E, D) ne donne aucune information sur les clés utilisés en cryptosystème.

Pour qu'un cryptosystème soit sûre, sa sécurité ne doit pas dépendre de ce qui ne peut pas être facilement changé, c'est-à-dire que la sécurité de cryptosystème doit ne pas être réduite à la connaissance de la clé, mais plutôt à son implémentation, son conception, et à sa transparence a la communauté cryptographique. Ce principe cryptographique est appelé le principe de Kerckhoff qui a été mis au point en 1883.

IV.3 Les différentes classes de chiffrement

La relation qui lie les deux clés $K\ et\ K'$ détermine la nature de cryptosystème, et par conséquent, l'algorithme de chiffrement correspondant. La communauté cryptographique ont défini deux classes d'algorithmes de chiffrement : algorithme de chiffrement symétrique et algorithme de chiffrement asymétrique.

IV.4 Le chiffrement symétrique

Le principe de chiffrement symétrique (ou à clé symétrique) consiste à chiffrer et déchiffrer le message en employant la même clé (voir figure IV.2), c'est-à-dire $K = K'$. Le chiffrement consiste alors à effectuer une opération entre la clé privée K et les données à chiffrer. Le déchiffrement se fait à l'aide de cette même clé secrète. Cette dernière est choisie aléatoirement à partir d'un espace de clés. Parmi les algorithmes les plus répandus, on peut citer DES [48] qui utilise une clé de 56 bits, AES [49] dont la taille de clé varie entre 128, 192, et 256, et 3DES [50] qui est une amélioration de son prédécesseur DES. L'avantage principal de ce type de chiffrement réside dans sa rapidité. Cependant, son inconvénient majeur se présente lors de la distribution de la clé, parce que le chiffrement symétrique préfère l'échange manuel de la clé communiquée K.

Figure IV.2 Le chiffrement symétrique.

IV.5 Le chiffrement asymétrique

Le chiffrement asymétrique (ou a clé publique) quant a lui, utilise des clés différentes pour le chiffrement et le déchiffrement (voir figure IV.3). Le chiffrement se fait en utilisant une clé dite publique connue de tous, alors que le déchiffrement est effectué en utilisant une clé privé seulement pour le destinataire.

Lorsqu'un utilisateur désire envoyer un message à un autre utilisateur, il lui suffit de chiffrer le message à envoyer au moyen de la clé publique K de destinataire (qu'il trouvera par exemple dans un serveur de clés tel qu'un annuaire ou bien en signature d'un courrier électronique). Le destinataire sera en mesure de déchiffrer le message à l'aide de sa clé privée K'(qu'il est seul à connaître).

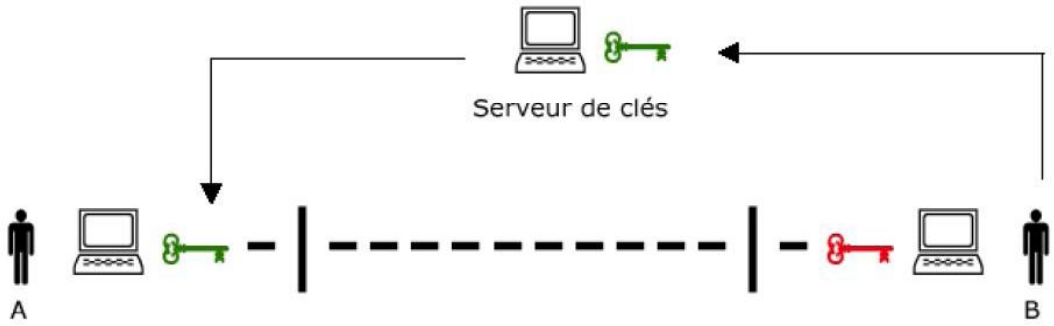


Figure IV.3 le chiffrement asymétrique.

Parmi les algorithmes de chiffrement asymétrique, on peut citer le RSA [51], ElGamal [52], ou Merkle-Hellman [53]. La taille des clés s'étend de 512 bits à 2048 bits en standard. Au niveau de performances, le chiffrement par voie asymétrique est environ 1000 fois plus lent que le chiffrement symétrique. Cependant, à l'inverse du chiffrement symétrique où le nombre de clés est le problème majeur, ici, seules n paires sont nécessaires. En effet, chaque utilisateur possède une paire (K, K') et tous les transferts de message ont lieu avec ces clés. La distribution de clés est grandement facilitée car l'échange de clés secrètes n'est plus nécessaire. Chaque utilisateur conserve sa clé secrète sans jamais la divulguer. Seule la clé publique devra être distribuée.

IV.6 Les modes d'opérations

Les modes d'opérations ou de chainage sont des solutions utilisées pour le chiffrement par blocs. Dans le cadre d'une implémentation pratique, l'algorithme ''pur'' est combinée à une série d'opérations simples en vue d'améliorer la sécurité sans pour autant pénaliser l'efficacité de l'algorithme. Cette combinaison est appelée un mode cryptographique opératoire [43].

Les messages à crypter peuvent avoir une longueur arbitraire. Pour adapter la taille de message à celle de la clef, on décompose le message par blocs de taille fixe correspondant aux tailles de clés que l'on chiffre ensuite un à un et que l'on envoie successivement. Pour cela quatre modes de chiffrement par blocs seront possibles : ECB, CBC, CFB, OFB et le mode CTR.

IV.6.1 Le mode ECB (Electronic Code Book)

Le mode ECB est le mode le plus aisé (voir figure IV.4). Le texte en clair M est patronné en série de n blocs $(m_i, i = 1, \dots, n)$ de taille fixe, et chaque bloc est chiffré indépendamment comme suit :

$$c_i = E_k(m_i), i = 1, \dots, n. \qquad \text{(IV.4)}$$

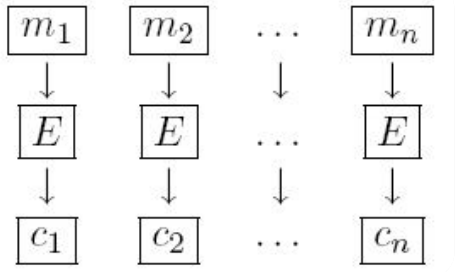


Figure IV.4 Le mode ECB

On construit le cryptogramme final par la concaténation de blocs chiffrés consécutivement.

Puisque les blocs m_i sont chiffrés de la même manière, ceci nuit la sécurité de cryptosystème. De plus, un attaquant intrus peut permuter entre deux blocs successifs sans que le destinataire s'aperçoive.

IV.6.2 Le mode CBC (Cipher Block Chaining)

Ce mode a été introduit pour pallier les inconvénients de mode ECB. Pour cela, après le découpage de M en série de blocs $(m_i, i = 1, \dots, n)$, chaque bloc sera chiffré de la manière suivante. Premièrement, un bloc d'initialisation c_0 est choisi. Après, chaque bloc clair m_i subit un XOR entre lui et le bloc chiffré c_{i-1} qui le précède. Finalement, le résultat de XOR est chiffré comme est illustré sur la figure IV.5

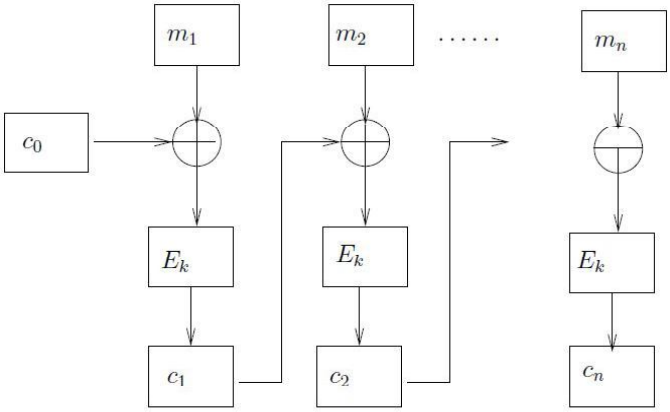


Figure IV.5 Le mode CBC

Ce type de chiffrement est utilisé dans la majorité d'applications contrairement à l'ECB qui n'est plus utilisé en pratique. Cependant, sa sureté dépend étroitement de la connaissance de la fonction inverse de $E_K$et de D_k.

IV.6.3 Le mode CFB (Cipher FeedBack)

Ce mode a été introduit afin d'éviter de voir le calcul de la fonction inverse de E_K ce qui permet en revanche de rectifier le problème de CBC. Chaque bloc clair $m_i$est XORé avec le résultat de chiffrement de bloc crypté c_{i-1}suivant le schéma présenté dans la figure IV.6.

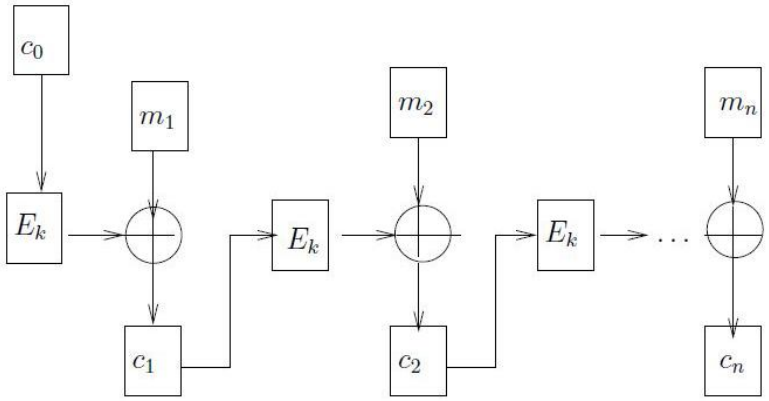


Figure IV.6 le mode CFB

IV.6.4 Le mode OFB (Output FeedBack)

Ce mode (voir la figure IV.7) est une variante de CFB qui permet d'avoir un chiffrement et un déchiffrement totalement symétrique.

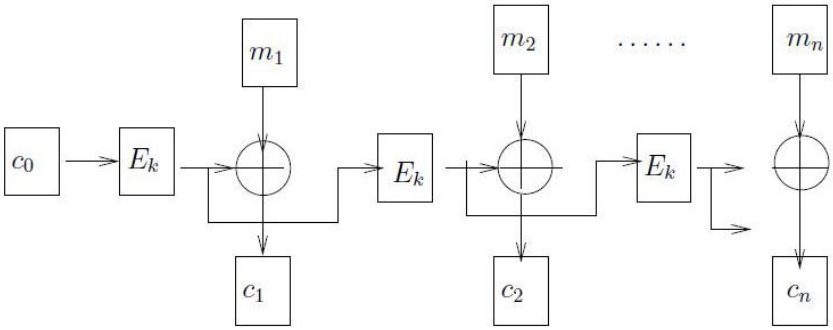


Figure IV.7 Le mode OFB.

IV.6.5 Le mode CTR (Counter-mode encryption)

Ce mode de chiffrement est lui aussi totalement symétrique et il est illustré dans la figure 4.8. Il utilise pour chiffrement un compteur de valeur initiale *T* avec :

$$c_i = m_i \oplus E_K(T + i) \qquad \text{(IV.5)}$$

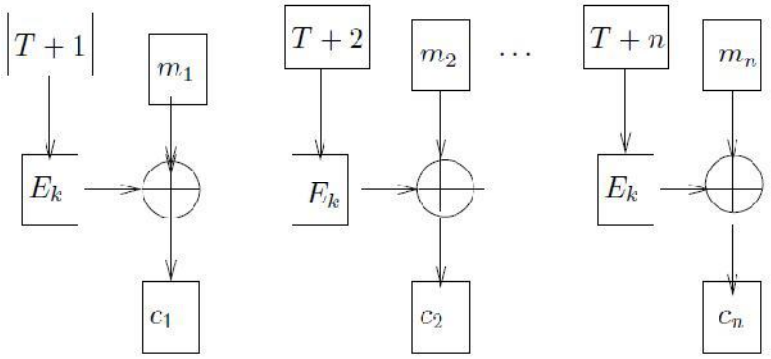


Figure IV.8 Le mode CTB.

IV.7 Les différents types d'attaques

La robustesse d'un cryptosystème contre des attaques malveillantes est un objectif principal en cryptographie. L'adversaire tente toujours à casser l'algorithme afin de restituer le texte en clair et/ou la clé de déchiffrement. Parmi les attaques malveillantes connues en cryptanalyse, on trouve :

a) attaque par force brute (brute-force attack) : l'adversaire tente de casser le cryptosystème en essayant toutes les clés possibles.
b) attaque à texte chiffré connu (Ciphertext-only attack) : l'adversaire ne connait que le texte chiffré.

c) attaque à texte clair connu (Known-plaintext attack) : l'adversaire dispose de texte clair et son texte chiffré correspondant.
d) attaque à texte clair choisi (Chosen-plaintext attack) : l'adversaire peut avoir une partie de texte claire a partir de celle de texte chiffré.
e) attaque à texte chiffré choisi (Chosen-ciphertext attack): l'adversaire peut trouver une partie de texte chiffré choisie a partir d'une partie d'un texte clair.

IV.8 Introduction a la sécurité d'images et de vidéos

De nos jours, la vidéo numérique joue un rôle crucial dans notre vie quotidienne grâce à sa mise en œuvre dans différentes applications comme VOD (Vidéo à al demande), la télévision numérique, diffusion en continu (streaming), vidéoconférence, vidéosurveillance. La variété massive d'applications qui utilise la vidéo nécessite la sécurité de la vidéo employée au moyen de techniques de protection qui garantissent une haute confidentialité.

La protection de l'information vidéo diffère de celle appliquée aux différentes formes de multimédia. Tout d'abords, les techniques classiques de cryptographie comme DES ou AES peuvent être appliquées efficacement aux fichiers de type texte, car on peut considérer celui-ci comme une suite de bits qui respecte un format de codage bien établi, par exemple le système de codage ASCII. A contrario, la protection de l'image ou la vidéo numérique au moyen de techniques de chiffrement classiques utilisées pour le texte peuvent altérer le format de codage ce qui conduit aux erreurs lors de décodage de fichiers cryptés. Ce qui implique la conception de techniques de chiffrement approprié pour chaque format de l'image.

Depuis son entré a l'ère numérique, l'état de l'art de chiffrement de l'image numérique [44] est enrichi exponentiellement par plusieurs techniques. Ces techniques peuvent s'appliquer avant la compression, durant la compression, et après la compression. La première catégorie consiste à trouver une approche de chiffrement de contenu visuel en mode spatial ou fréquentiel. Ce mode affecte la taille d'image considérablement en offrant en revanche un haut niveau de sécurité. La deuxième catégorie consiste à chiffrer l'image durant sa compression pour atteindre trois objectifs essentiels : le premier est de dégrader le contenu visuel suffisamment ;

deuxièmement, à ne pas affecter la taille de l'image claire, c'est-à-dire qu'il faut que la taille de l'image claire soit la même que celle de l'image cryptée, et dernièrement, est que le format de l'image cryptée soit conforme au standard de codage. La dernière catégorie de chiffrement d'image consiste à chiffrer le fichier d'image entièrement par des techniques de chiffrement classiques au détriment de la conformité de format de codage de l'image cryptée. En conséquence, l'image cryptée ne sera pas décodable.

La protection de la vidéo numérique est un axe de recherche qui dépend fortement de la compression et de codage. Comme la vidéo est une séquence d'images consécutives, aucun changement affecte un bloc d'une image donnée, il sera propagé aux images consécutives lors de décodage de vidéo. Le chiffrement consiste alors à dégrader autant que possible le contenu informatif (visuel ou binaire) de la vidéo. L'application de chiffrement visuel sur chaque image séparément conduit inévitablement a un changement de volume de vidéo cryptée. Aussi, le chiffrement de flux binaire de la vidéo compressée altère le format de vidéo ce qui génère un fichier non décodable, et par conséquent, le destinataire ne peut pas voir le contenu de la vidéo. Le chiffrement appliqué conjointement durant la compression est la seule solution permettant d'avoir une vidéo cryptée conforme à la norme avec une taille aussi proche de la vidéo claire, avec un niveau de sécurité visuelle suffisante.

IV.9 Les différente classes de chiffrement d'images et de vidéos

IV.9.1 Le chiffrement total (full encryption)

Ce type de chiffrement consiste à encrypter toutes les données de l'information claire. Ces données sont très variées et peuvent être des données spatiales comme en [45], des données fréquentielles comme en [46], ou de flux binaire de fichier compressé comme en [47]. Dans ce dernier cas, le flux binaire crypté ne sera pas conforme a la norme de flux en clair, les éléments syntaxiques après le chiffrement ne seront pas lisibles par le décodeur.

Ce type de chiffrement qui est beaucoup utilisé pour la sécurité d'images, influe sur la taille de l'image cryptée. A contrario, il est rarement employé pour le chiffrement de vidéo, car s'il est appliqué pour chiffrer chaque image séparément de la séquence vidéo en mode spatial/fréquentiel, il conduira à accroitre le volume de vidéo chiffrée.

De même, s'il est appliqué pour chiffrer le flux binaire compressé, il va endommager son format. Grace à ces raisons, il est déconseillé de protéger la vidéo au moyen de ce type de chiffrement

IV.9.2 Le chiffrement sélectif (selective encryption)

Contrairement au chiffrement total, le chiffrement sélectif tente à chiffrer seulement un sous ensemble des données de l'image ou la vidéo à crypter. Les données chiffrées sont sélectionnée selon des critères et des conditions très variés. Mais le plus souvent, Les critères de sélection sont juste des conditions qui garantissent la confidentialité et la conformité de format de fichier compressé.

Le chiffrement sélectif est appliqué souvent durant l'étape de compression afin d'obtenir un fichier conforme a la norme avec une taille proche ou identique au fichier clair, avec un haut niveau de sécurité.

IV.9.3 Le chiffrement transparent

L'objectif primordial de chiffrement transparent est de réduire la qualité visuelle de l'image/vidéo de tel sorte que le contenu visuel sera lisible après le décodage mais avec une mauvaise qualité.

IV.10 Evaluation de performances de techniques de chiffrement de vidéos numériques

Pour mieux analyser quantitativement les techniques de cryptage vidéo, il est nécessaire de définir un ensemble de paramètres ou indices de performance considérés comme des métriques pour l'évaluation d'un système de chiffrement de vidéo. Ces mesures permettent à l'utilisateur d'évaluer la performance d'un algorithme de chiffrement vidéo. Parmi ces mesures, on trouve :

- L'efficacité computationnelle : elle peut être définie en termes de complexité spatiale et de complexité temporelle. La complexité spatiale est déterminée par l'exigence de mémoire (RAM, ROM, registres,…) pour la sauvegarde de code de cryptosystème et ses données associées. La complexité temporelle est déterminée par la mesure de temps de calcul nécessaire requis pour

accomplir le chiffrement/déchiffrement en fonction de la complexité logicielle/hardware offerte pour son implémentation.

- Le niveau de sécurité proposée : Chaque application vidéo nécessite des exigences relatives à la sécurité. Les applications de VoD exigent une sécurité visuelle très confidentielle. Les applications de vidéoconférence ou de téléphonie vidéo (les appels vidéo par exemple) exigent une communication fermée pour tous les autres utilisateurs non concernés par la conversation.
- L'efficacité de compression : Les données vidéos sont généralement très grande, et en général on utilise des données compressées afin de réduire l'espace de stockage et d'économiser la bande passante. Lors de la conception d'un algorithme de chiffrement, il est de préoccupation majeure que la taille de la vidéo comprimé ne doit pas être augmentée après le cryptage.
- La conformité/compliance/portabilité de format de la vidéo cryptée : le chiffrement ne doit pas altérer le format de flux binaire de la vidéo compressée, car le flux n'est qu'une suite d'éléments syntaxiques codés en binaire, et décodables selon une sémantique et une syntaxe bien établie. Une petite modification inappropriée de ceux-ci altère la lecture de flux binaire, et par conséquent, il devient non décodable.
- La sécurité visuelle : Elle est liée au niveau de la dégradation visuelle perçue après le chiffrement. Certaines approches effacent entièrement le contenu visuel. Par contre, d'autres approches choisissent la réduction de la qualité visuelle par des techniques de chiffrement transparent.

IV.11 Etat de l'art de chiffrement de vidéos numériques

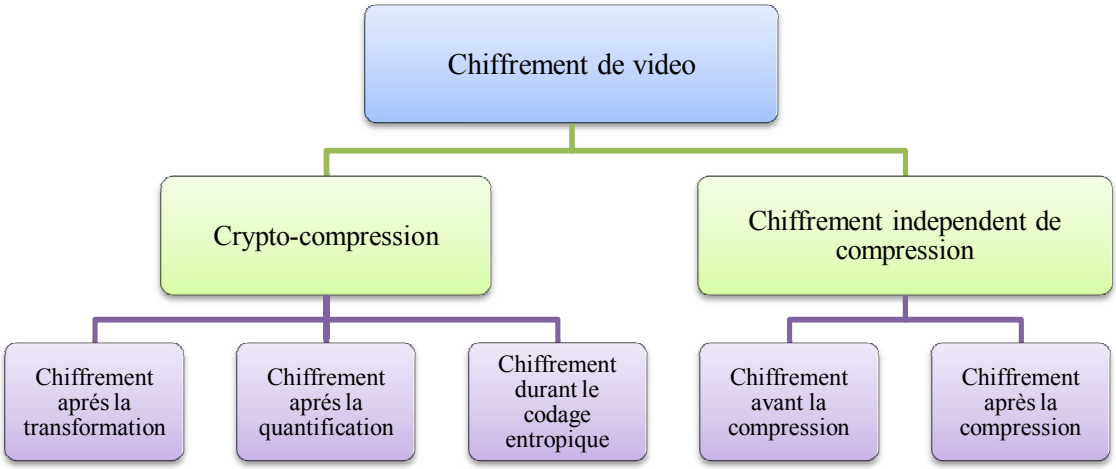


Figure IV.9 Taxonomie des techniques de chiffrement de vidéo numérique.

Depuis les années 90s, les approches de chiffrement de vidéos numériques accroissent exponentiellement après chaque sortie d'une nouvelle norme de compression vidéo, surtout celles standardisées par les deux communautés de normalisation ITU-T et IEC. La bibliographie scientifique est enrichie dans cette dernière décennie par des approches de chiffrement de H.264, car elle offre des notions nouvelles et efficaces pour la compression. En plus, les défis pour le chiffrement sélectif durant le codage entropique était toujours un thème de recherche très actif.

L'univers d'approches de chiffrement de vidéo est devisé suivant la taxonomie présentée dans la figure IV.9. Premièrement, la relation entre la compression et le chiffrement nous amène à définir deux classes primaires : des approches de chiffrement qui s'effectuent conjointement durant la compression ou ce qu'on appelle les systèmes de crypto-compression, et des approches de chiffrement qui ne dépendent pas d'aucune étape de compression vidéo. On va donner dans la prochaine section, les approches les plus populaires et émergentes pour chaque classe primaire de chiffrement de vidéos numériques.

IV.11.1 Le chiffrement indépendant de la compression

IV.11.1.1 Avant la compression

Les algorithmes de compression ont l'intention de réduire autant que possible la redondance de plaintext d'entrée. Les algorithmes de chiffrement cachent la redondance inhérente de plaintext d'entrée qui utilise des opérations cryptographiques. En conséquence, il y a beaucoup moins de redondance pour comprimer si ces algorithmes de chiffrement sont placés avant compression. Par conséquent, les algorithmes du chiffrement sont rarement rendus effectifs avant compression. Parmi les algorithmes de chiffrement se plaçant avant la compression, on trouve : l'approche de Pazarci-Dipc [54], et l'approche de chiffrement a base de préservation de corrélation de vidéo CPEV[55] (correlation-preserving encryption video).

L'approche de Pazarci-Dipcin consiste à chiffrer la vidéo dans l'espace de couleur RGB, et en utilisant quatre transformations linéaires secrètes avant la compression de la vidéo. Elle permet en outre, de garder la même taille de vidéo originale. Cependant, [56] a prouvé que cette approche n'est pas robuste contre l'attaque a force brute, car la taille de l'espace de clés n'est plus suffisamment large.

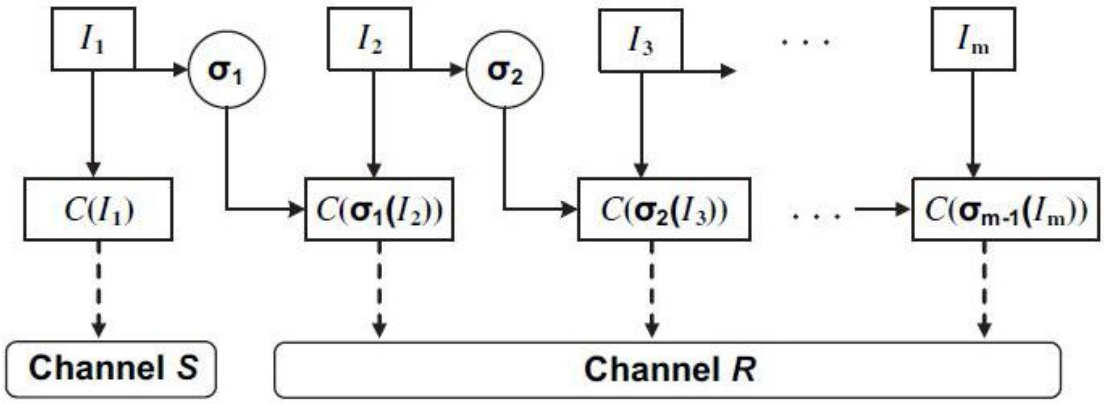


Figure IV.10 Le schéma de CPEV [55].

L'idée de base de CPEV est de calculer une image de tri σ_ia partir de l'image i qui aide à permuter les pixels de l'image suivante $i+1$ comme il est illustré dans la figure IV.10. Après le calcul de σ_1, la première image de la séquence vidéo est comprimée et transmise au destinataire. Après, les pixels de la deuxième image sont permutés dont les positions sont indiqués dans σ_1, ensuite, l'image résultante de la

permutation est compressée. Le même processus de chiffrement de la deuxième image sera appliqué aux autres images de la séquence.

IV.11.1.2 Après la compression

Le chiffrement naïf de la vidéo compressée peut altérer le décodage de la vidéo cryptée, car le format de flux binaire ne sera pas conforme avec la norme de codage convenue. Certaines approches comme SECMPEG [57] et VEA [58] ont réussi à chiffrer le flux binaire tout en préservant son format pour le décodage.

Meyer et Gadegast [57] ont proposé un chiffrement sélectif pour la norme MPEG1 en 1995. Les parties sélectionnées pour la protection sont chiffrées par des algorithmes de chiffrement conventionnels. Selon la quantité de données à être cryptée, quatre niveaux de sécurité sont définis :

- premier niveau qui s'applique pour le chiffrement d'entêtes de la couche de la séquence (sequence layer), et les entêtes des couches de tranches.
- deuxième niveau qui permet de chiffrer les coefficients de DCT de basse fréquence de chaque bloc dans chaque image intra.
- troisième niveau qui permet de chiffrer seulement les blocs intra.
- quatrième niveau qui permet de chiffrer entièrement le flux binaire de la vidéo compressée.

Qiao et Nahrstedt [58] ont introduit un algorithme de chiffrement de MPEG qui s'appelle VEA en 1998. Il est basé sur une analyse statistique sur les valeurs d'octets constituantes le flux binaire de la vidéo comprimée. L'idée de VEA est simple : tout d'abords, une première étape est de localiser les octets qui suivent une loi uniforme dans le flux binaire. Ces octets sont chiffrés par des algorithmes de chiffrements conventionnels. Les octets chiffrés sont utilisées comme des clés pour chiffrer le reste d'octets avec l'opération de XOR.

Si MPEG permet l'émergence de SECMPEG et VEA, H.264 n'est plus le cas. ABOMHARA et al. [47] ont proposé un algorithme de chiffrement de flux binaire compressé en H.264 en employant l'AES. Malheureusement, le format de flux binaire crypté n'est pas conforme à la syntaxe approuvée de la norme H.264.

IV.11.2 Les systèmes de crypto-compression pour la sécurité de vidéos

L'intégration de module de chiffrement durant la compression et le codage de la vidéo numérique est un thème de recherche très actif en recherche scientifique. D'abords, il permet de préserver le format de flux binaire crypté avec une taille proche de celle de flux original. Le chiffrement peut être s'appliquer a n'importe quel stage de compression : après la transformation fréquentielle, après la quantification visuelle, et durant le module de codage entropique.

IV.11.2.1 Apres la transformation

Les données à chiffrer après la transformation fréquentielle de l'erreur résiduelle sont les amplitudes et les signes des coefficients. Chaque norme de codage dispose de son propre transformée appliquée. Le plus populaire est la transformée de DCT et ses amélioration.

Zeng and Lei [59] ont proposé une approche de chiffrement sélectif appliquée dans le domaine fréquentielle. Premièrement, les images intra sont devisées en segments qui sont composée de macroblocs et/ou blocs. Dans chaque segment, les coefficients DCT de même groupe (emplacement de fréquence) sont mélangés de façon aléatoire à l'intérieur d'un segment selon une table de permutation commandée par la clé de cryptage. Le bit de signe de coefficient DCT détermine si sa valeur est positive ou négative. Le chiffrement de signes est très efficace pour la dégradation de qualité visuelle d'image intra. Pour un haut niveau de sécurité, les bits de signes des coefficients de DCT sont déplacés aléatoirement. Deuxièmement, quant aux images prédites ou bi-prédites, les bits de signes de vecteurs de mouvements sont aussi chiffrés. De plus, les signes et les amplitudes des coefficients sont chiffrés de la même manière comme est appliqué dans les images intra.

Le chiffrement des coefficients de DCT est moins préféré, car les amplitudes seront modifiées après la quantification qui est une fonction irréversible ; les amplitudes reconstruites après la quantification inverse diffèrent entièrement de celles non quantifiées car la quantification est irréversible.

IV.11.2.2 Après la quantification

La quantification est l'étape qui permet la réduction de l'espace de coefficients. Ces derniers seront balayés et parcourus selon un mode de balayage. Le mode en zigzag est le plus populaire car il commence par les coefficients de basses fréquences, et il termine par les coefficients à hautes fréquences dont l'ordre de chaque QTC est déterminé selon la norme de codage adoptée. Les données possibles à chiffrer sont : les amplitudes et les signes de QTCs, et aussi l'ordre de QTCs.

Tang [60] a proposé en 1996 un algorithme de permutation en zigzag appliqué a chaque bloc de 8×8 QTCs. L'idée fondamentale de cet algorithme est qu'il lit les 64 QTCs rapidement en utilisant une liste de permutation choisie au hasard au lieu de l'ordre en zigzag ordinaire pour chaque bloc quantifié. L'algorithme se compose de trois étapes :

1) Une liste de permutation de cardinal 64 est générée aléatoirement en mode hors ligne (offline).
2) Les DCs de tous les blocs sont concaténés dans une seule liste et sont chiffrés avec l'algorithme DES. Les huit bits de chaque DC encrypté notés $d_7d_6d_5d_4d_3d_2d_1d_0$ sont devisé en deux listes $d_7d_6d_5d_4$ et $d_3d_2d_1d_0$ respectivement. La première liste remplace les coefficients de DCs originaux et la deuxième liste remplace le dernier coefficient de AC de bloc (AC_{63}). Cette procédure est utilisée pour éviter que le coefficient de DC puisse être identifié, car son amplitude est souvent la plus grande parmi tous les coefficients de bloc.
3) La liste de permutation générée est appliquée pour changer les positions des QTCs dont l'ordre est mentionné dans la liste.

Shi et al. [61] ont proposé en 1999 un algorithme qui s'appelle RVEA (Real-time video encryption algorithm), dont l'idée de base est de chiffrer seulement une portion de QTCs afin de réduire le temps de calcul. L'avantage principal est qu'il puisse choisir seulement 10% de flux binaire à crypter en laissant le reste en clair, ce qui permet en revanche de faciliter son implémentation en temps réel. Malheureusement, bien que cet algorithme résiste contre une attaque à texte chiffré connu, RVEA est vulnérable contre des attaques perceptuelles comme il est montré dans [62]. PVEA

(Perceptual Video Encryption Algorithm) [63] est une amélioration de RVEA proposée par Li et al. Elle consiste a renforcer RVEA par choisir seulement des codes a longueur fixes FLCs (fixed-length code) a partir de flux de bits de MPEG, et ceci est atteint selon des paramètres de contrôle de sécurité visuelle souhaitée.

IV.11.2.3 Durant le codage entropique

Après la sortie et la standardisation de chaque norme de codage vidéo, la préservation de la taille de flux binaire crypté sans augmentation, et avoir un format conforme décodable selon la syntaxe de la norme, occupe une préoccupation majeure pour la communauté cryptographique, car elle représente un défi réel à surmonter. Elle repose sur l'étude de la décodabilité des éléments syntaxiques après le chiffrement. Le choix de ces éléments syntaxiques et sa protection par un chiffrement sélectif sont des étapes communes suivies par la majorité d'approches de chiffrement durant le codage entropique existantes dans la littérature scientifique.

Les données à crypter varient selon le codage entropique adopté par la norme de codage vidéo. Les normes de MPEG utilisent quant à eux, des tables de Huffman, où dans ce cas, chaque table se compose en codes de type VLC. H.264 et ses extensions utilisent un codage adaptatif selon le contexte par l'emploi de CAVLC et CABAC.

Parmi les approches existantes pour le chiffrement de tables de Huffman, on trouve l'approche MHT (Multiple Huffman Tables) [64]. L'idée de base de cette approche est de transformer le codeur entropique en un système de chiffrement (voir figure IV.11). Ceci est accompli en utilisant des modèles statistiques multiples qui remplacent ultérieurement le seul modèle statistique employé par l'encodeur vidéo. Le choix de tables de Huffman particulières et l'ordre comme elles sont utilisées, est maintenu secret comme la clef de l'algorithme de chiffrement. Au lieu de chiffrer les données d'entrée directement, MHT utilise des tables de Huffman pré-entreposées présentes pour chiffrer les coefficients DCT quantifiés. Ces tables sont différentes de celles utilisées depuis le codeur entropique de MPEG, et son emploi permet d'effacer le contenu visuel de la vidéo après le chiffrement. L'avantage remarquable de MHT est le cout calculatoire très réduit en comparaison avec les autres algorithmes de chiffrement de MPEG.

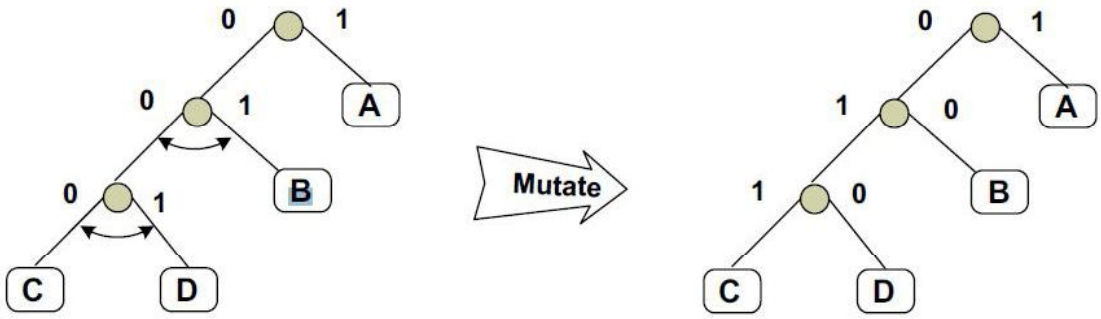


Figure IV.11 Un exemple d'un chiffrement de MPEG en utilisant l'approche de MHT [64].

La dernière décennie s'est caractérisée par l'émergence de techniques de chiffrement de H.264 qui emploi deux codeurs entropiques CAVLC et CABAC. L'inclusion de module de chiffrement durant le codage entropique permet d'avoir un flux vidéo crypté conforme à la syntaxe de la norme ayant une taille sans augmentation. Le point crucial est de localiser les éléments syntaxiques à crypter qui n'empêchent te n'interrompre pas le processus de décodage après leur modification.

Parmi les éléments syntaxiques de CAVLC, on peut citer :

- Les signes de MVD.
- Les signes de T1s (élément syntaxique qui indiquent si le QTC est égal a 1).
- Le suffixe de EG_0 d'amplitude de QTC non nul.
- Le signe de QTC non nul.

J. Wang et al. [65] ont présenté dans ses travaux une approche de chiffrement sélectif qui protège les signes de MVD et les signes de T1s au moyen de XOR. D. Wang et al. [66] ont choisi à chiffrer les éléments syntaxiques de modes intra et ceux de signes de T1s en utilisant RC4 [67]. H. Sohn et al.[68] ont choisi a chiffrer seulement les signes de QTCs non nul.

De même, on trouve dans CABAC, ces éléments qui peuvent être cryptés :

- le suffixe de code résultant de la concaténation de code unaire et EG_3 noté aussi par U/EG_3.
- Le suffixe de code résultant de la concaténation de code unaire et EG_0 noté aussi par U/EG_0.

- Signes d'amplitude des éléments syntaxiques NZ-TCs qui servent pour représenter les signes des QTCs non nuls.

Les éléments syntaxiques présentés ci-dessus sont obligatoirement chiffrée avant leur codage arithmétique binaire.

H.-J. Lee et al. [69] dans leur approche, ont chiffré tous les éléments syntaxiques qui représentent les QTCs non nuls au moyen d'un XOR avec une séquence pseudo-aléatoire. Y. Kim et al. [70] ont présenté une autre approche qui permet de chiffrer les signes d'amplitude des éléments syntaxiques NZ-TCs et certain éléments syntaxiques de CABAC avec un XOR entre un plaintext et une clé. Z. Shahid et al. [71] dans leur publication, ont présenté une approche de chiffrement sélectif qui permet de chiffrer les signes de T1s, les signes de QTCs non nuls, les suffixes de EG_0 et les signes de QTCs non nuls pour CABAC. Dans leur approche, les éléments syntaxiques sont chiffrés au moyen du cryptosystème AES en mode opérationnel CFB.

N. Asghar et al. [72] ont étudié en 2014 dans leur publication l'ensemble de tous les éléments syntaxiques entropiques suffisantes pour le chiffrement de H.264 que ce soit dans CAVLC ou dans CABAC.

Après sa sortie a la communauté scientifique, le chiffrement de l'information video codée en norme HEVC a tenu beaucoup de considération. Puisque la video HEVC représente une grande quantité visuelle, le défi majeur était de trouver les éléments syntaxiques entropiques à chiffrer.

Z. Shahid et al. [73][74] dans un deuxième effort, ont tenté de chiffrer les codes utilisés depuis CABAC pour chiffrer les QTCs non nuls qui sont EG_0 et TR_p (code de Golomb-rice). Ils ont juste transposé leur approche de [71] pour chiffrer les codes EG_0 et TR_p selon le brouillon de travail de HEVC ver 6 (WD6 [75]). Malheureusement, le codage de QTCs a connu beaucoup d'améliorations dans le brouillon standardisé [37], et en conséquence, les approches [73] et [74] nécessitent des mises a niveau car le codage de QTCs dans H.264 diffère carrément de celui appliqué dans HEVC.

IV.12 Conclusion

Nous avons vu dans ce chapitre, une étude sur les différentes techniques de chiffrement de l'information vidéo. Elles se distinguent suivant la norme de codage, et aussi, selon le stade où elles s'appliquent.

Les techniques de chiffrement de HEVC de Z. Shahid et al. sont devenue non appropriées car elles sont proposés avant Avril 2013, la date de standardisation de HEVC où le codage des données fréquentielles est totalement changées dans le document standardisé de HEVC. Ce qui implique en revanche de trouver une approche alternative afin de surmonter le défi de trouver un cryptosystème durant le codage entropique, et qui permet en outre de protéger les données fréquentielles sans altérer la conformité de flux binaire crypté. C'est le but visé dans les prochains chapitres.

Chapitre V.

Un chiffrement sélectif robuste et rapide pour la protection de vidéo HEVC

V.1 Introduction

Après sa standardisation, la norme HEVC est attendue à remplacer son prédécesseur H.264 dans tous les domaines d'applications numériques et de recherche. La protection de flux binaire à l'aide des techniques de cryptographie sera sans doute l'une des préoccupations émergentes pour la recherche scientifique. HEVC est une solution conçue pour coder des vidéos à haute définition (HD, Ultra HD, 2K, 4K, 8K,…), c'est-à-dire des données a quantité volumineuse énorme. Les approches de chiffrement s'appliquant avant la compression augmentent la taille de la vidéo compressée. Et les approches s'achevant après la compression vont altérer sans doute le format de la vidéo compressée. Par conséquent, la conception d'un système de crypto-compression qui se procède durant le codage entropique s'avère la seule solution permettant ainsi de préserver la taille et le format de la vidéo compressée.

CABAC est le seul codeur entropique adopté par la norme HEVC. Il permet de transformer des éléments syntaxiques en une suite de bits. Les données entropiques relatives au codage des données fréquentielles (surtout les QTCs) permettent sans doute de contrôler la qualité visuelle de la vidéo compressée. En conséquence, La protection des ses données permet en revanche de résulter une vidéo compressée cryptée avec une haute confidentialité visuelle.

L'objectif de ce chapitre est de présenter une nouvelle approche de chiffrement sélectif conforme à la norme HEVC [76]. Elle permet en outre de générer un flux binaire décodable selon la dernière version de document standardisé de HEVC. Notre approche consiste en une approche de chiffrement sélectif par la sélection des signes et des codes de type Golomb-Rice des QTCs non nuls, et les chiffrer a l'aide d'un cryptosystème AES en mode opératoire CBC. Après avoir présenté le codage entropiques des QTCs selon les dernières modifications achevées en HEVC, nous allons exposer la problématique de chiffrement par mentionner les inconvénients et

les lacunes observées dans les travaux antérieurs. Par la suite, nous allons expliquer en détail notre approche proposée. La validation de notre approche est évaluée par l'illustration de plusieurs résultats expérimentaux. Finalement, Nous allons clôturer notre chapitre par une conclusion.

V.2 Le codage entropique de QTCs en HEVC

Le codage entropique en HEVC est l'un des champs de recherche qui a pris largement de temps avant sa normalisation. Effectivement, Il est passé par plusieurs modifications en commençant par la proposition de CAVLC et de CABAC dans les premières versions de brouillons de travails (WD), l'annulation de CAVLC à partir de WD ver 6 [75]. CABAC devient en conséquence le seul codeur à adopter. L'optimisation de codage binaire des données fréquentielles a occupé quant à elle leur part considérable de recherche.

Après leur transformation et quantification de l'erreur résiduelle à l'aide de RQT, les blocs d'unité de transformé ayant une taille supérieur a 4×4 seront transformé en blocs de 4×4 QTCs, où chaque bloc contient 16 QTCs qui seront par la suite rangés en un vecteur dont l'ordre de chaque QTC est obtenu selon un parcours diagonal inverse en zigzag. .

Le codage de QTCs est assuré par un algorithme de codage qui emploie cinq éléments syntaxiques qui sont :

a) significant_coeff_flag : qui indique si le coefficient QTC est nul ou non. S'il est nul, alors il va prendre la valeur ''0'' sinon il va prendre la valeur ''1''. Les significant_coeff_flag de tous les QTCs forment une chaine binaire qui s'appelle la carte de signifiance (significance map).

b) coeff_sign_flag : il représente le signe de QTC non nul. Si le QTC est négatif, alors coeff_sign_flag prend la valeur ''0'', sinon il va prendre la valeur ''1''.

c) coeff_abs_level_greater1_flag: indique si l'amplitude de QTC est supérieur a un : s'il est supérieur à un, alors il va prendre la valeur ''1'', autrement il va prendre la valeur ''0''.

d) coeff_abs_level_greater2_flag : indique si l'amplitude de QTC est supérieure à deux : s'il est supérieur à deux, alors il va prendre la valeur ''1'', autrement il va prendre la valeur ''0''.

e) coeff_abs_level_remaining : est utilisé pour coder les amplitudes de QTCs restants. Et en général, il est employé pour les QTCs ayant $|QTC| \geq 2$.

Les éléments syntaxiques coeff_abs_level_greater1_flag, coeff_abs_level_greater1_flag, et significant_coeff_flag sont codé en mode régulier de CABAC, c'est-à-dire ils nécessitent des tables de contextes pour leur codage. Cependant, les éléments syntaxiques restants qui sont coeff_abs_level_remaining et coeff_sign_flag n'exigent plus de tables de contexte, et ils sont codés en mode bypass qui assume 0.5 de probabilité pour le codage de chaque séquence binaire. La figure V.1 illustre un bloc de 4×4 coefficients quantifiés, l'ordre de parcours, et les valeurs données calculées pour ces éléments syntaxiques.

18	6	-6	-1
-12	4	-4	0
7	4	2	1
2	4	-1	0

Un bloc de 4×4 QTCs

16	14	11	7
15	12	8	4
13	9	5	2
10	6	3	1

L'ordre de parcours

	0	1	-1	0	2	4	-1	-4	4	2	-6	4	7	6	-12	18
significant_coeff_flag	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1
coeff_abs_level_greater1_flag		0	0		1	1	0	1	1	1						
coeff_abs_level_greater2_flag					0											
coeff_sign_flag		0	1		0	0	1	1	0	0	1	0	0	0	1	0
coeff_abs_level_remaining						2		2	2	0	5	3	6	5	11	17

Figure V.1 Exemple de calcul d'éléments syntaxique pour un bloc de 4×4 QTCs.

Tous ces éléments syntaxiques ne passent pas par l'étape de binarisation excepté l'élément syntaxique coeff_abs_level_remaining qui nécessite cette étape car il représente une valeur entière.

La figure V.2 montre la binarisation de coeff_abs_level_remaining selon les dernières modifications achevées dans le document standardisé de HEVC [37, pp. 180-181]. Le code binaire de coeff_abs_level_remaining est une séquence de

décisions binaires appelés ***les bins***, et est le résultat de concaténation de deux chaines préfixe et suffixe. Cette binarisation dépend de deux paramètres qui sont : cTRMax et cRiceParam. cRiceParam peut prendre une valeur allant de 0 jusqu'à 4, et s'est déterminé en fonction d'amplitudes précédemment codées.

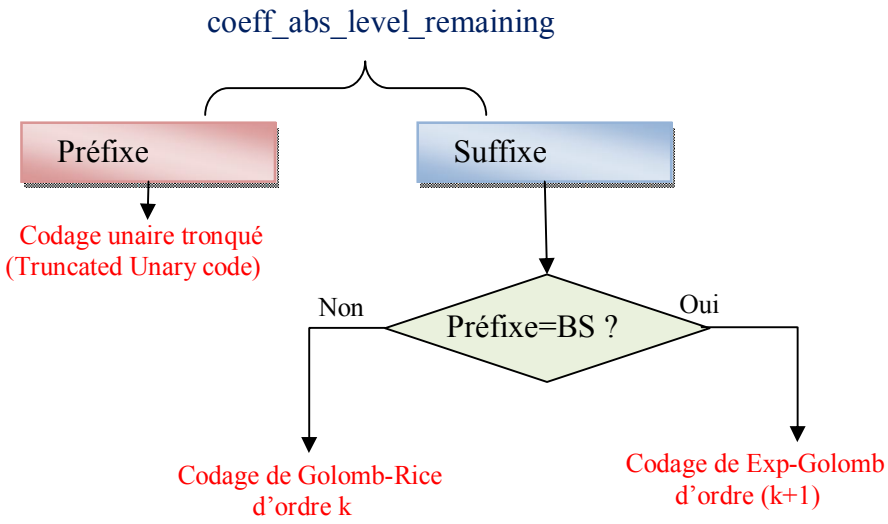


Figure V.2 La binarisation de coeff_abs_level_remaining selon [37]

Premièrement, la chaine de préfixe est calculé en utilisant le code unaire tronqué de la partie Min(4, coeff_abs_level_remaining >> cRiceParam). Si le préfixe est égal en représentation binaire a une chaine BS, alors le suffixe s'obtient en calculant le suffixe du code de Exp-Golomb d'ordre (cRiceParam+1) de la partie (coeff_abs_level_remaining - cTRMax). Autrement, le suffixe s'obtient en calculant le suffixe de code de Golomb-Rice de coeff_abs_level_remaining avec un ordre égal a cRiceParam, et est calculé par les ''cRiceParam'' bits de la représentation binaire de la partie (coeff_abs_level_remaining- ((coeff_abs_level_remaining>> cRiceParam) << cRiceParam). Finalement, on note que le suffixe n'existe pas pour les codes de Golomb-Rice ayant le paramètre cRiceParam nul.

HEVC utilise des codes de Golomb-Rice pour coder les QTCs ayant faibles amplitudes, et utilise des codes de Exp-Golomb pour coder les QTCs ayant forts amplitudes.

V.3 Problématique

Dés le lancement de projet HEVC en 2010, le chiffrement durant le module de CABAC a constitué l'un des défis à surmonter pour la communauté scientifique. Plusieurs travaux ont été proposés, parmi eux, on peut citer ceux de Z. Shahid et al. [73][74], de Wallendael et al. [77], et de Hofbauer et al.[78] qui ont été les premières solutions proposées pour la sécurité de HEVC.

En [78], Hofbauer et al. ont proposé une approche de chiffrement transparent par la protection d'un ensemble de signes de ACs. Cependant, ce type de chiffrement ne permet pas d'effacer le contenu visuel de fichier crypté. Wallendael et al. ont proposé en [77] la protection un ensemble d'éléments syntaxiques qui appartient a l'unité de codage, a l'unité de prédiction, et a l'unité de transformée. Le but de cette étude est de valider l'applicabilité de chiffrement sur ces éléments syntaxiques selon les dernières nouveautés vues dans le standard HEVC. Malheureusement, son étude n'a pas comporté le chiffrement d'amplitudes de QTCs.

Z. Shahid et al. ont proposé en [73] et [74] de chiffrer les amplitudes et les signes de QTCs en utilisant l'AES en mode opérationnel CFB. Ils ont tenté dans leurs travaux de transposer leur approche antérieur concernant le chiffrement de H.264 [71] en exploitant la ressemblance qui existe entre les codes utilisés en HEVC (Golomb-Rice , Exp-Golomb) et ceux de H.264 (unaire, Exp-Golomb). Ces approches se sont achevées selon le brouillon de travail WD ver 6 [75], où coeff_abs_level_remaining est binarisé soit par les codes de Golomb-Rice, soit par les codes de Exp-Golomb d'ordre zero EG_0. Malheureusement, le codage de coeff_abs_level_remaining est changé totalement dans le document standardisé final de HEVC [37] : le calcul de cTRMax et de cRiceParam est rénové, aussi l'utilisation des codes de Exp-Golomb d'ordre zero est annulée, et aussi la ressemblance discutée en [73] et en [74] n'est pas une condition suffisante pour transposer [71].

Dans [74], Z. Shahid et al. ont considéré que seulement les codes de Exp-Golomb et les codes de Golomb-Rice ayant des longueurs multiples de $2^{cRiceParam}$ sont des codes chiffrables. Et pour chiffrer les autres codes inchiffrables, il faut les décomposer auparavant en codes chiffrables. Après, une modification ultérieure de modèles de contextes a été proposée. La décomposition d'un seul code de Golomb-Rice en sous

codes n'est pas permise dans le document standard de HEVC, car chaque code représente un seul QTC, et le nombre de QTCs non nuls à décoder est déjà fixé et limité auparavant dans la carte de signifiance. Alors, une décomposition d'un seul code en sous codes signifie que le décodeur va interrompre le processus de décodage car il va trouver sans doute des codes additionnels qui ne sont pas déclarés dans la carte de signifiance.

A la lumière de nouveautés achevées en [37], il est nécessaire de concevoir une nouvelle approche pour protéger les QTCs durant le codage entropique ; c'est le but de notre approche.

V.4 AES

En 1997, le NIST (National Institue of standards and Technology) américain lance un appel d'offre afin de trouver un remplaçant au DES. 15 algorithmes ont été étudiés en fonction des différents critères. En 2000, l'algorithme belge Rijndaël proposé par Joan Daemen et Vincent Rijmen est retenu. Il faut savoir que chacun des modèles a été testé sur plusieurs types de surface et Rijndaël n'a été le premier sur aucune d'elle, mais il a montré qu'il gardait à chaque fois des performances très intéressantes. Il a donc été choisi autant pour son adaptabilité que pour ses performances. De fait de son origine, l'AES (pour advanced encryption system) [79] est devenu un standard. Il est donc libre d'utilisation, sans restriction d'usage ni brevet.

L'AES opère sur des blocs rectangulaires de 4 lignes dont chaque éléments (appelé octet ou byte) est composé de 8 bits. La clé peut être d'une longueur de 128, 156, ou 256 bits, de même pour le message clair et le message chiffré. L'AES a été choisi pour être totalement sûr et opérationnel sur tout type d'environnement.

V.5 L'approche proposée

Les éléments syntaxiques qu'on peut chiffrer sont ceux qui ont été codé en mode bypass. Donc, parmi les cinq éléments cités dans les sections précédentes, on peut crypter seulement coeff_sign_flag et les suffixes de coeff_abs_level_remaining. Car le préfixe de coeff_abs_level_remaining n'est pas modifiable, et les autres éléments syntaxiques exigèrent des tables de contextes pour initialiser leur codage entropique.

Les codes de Golomb-Rice sont beaucoup utilisés pour coder (au sens de binarisation) les QTCs plus fréquents, par contre les codes de Exp-Golomb sont utilisé pour coder les QTCs moins fréquents. Dans notre approche, nous avons choisi à chiffrer seulement les codes de Golomb-Rice en déterminant ainsi les conditions de leur sélection.

La longueur de chaque suffixe de code de Golomb-Rice varie de un jusqu'à quatre. Par exemple si cRiceParam est égal à trois, alors le suffixe peut être : 000, 001, 010, 011, 100, 101,110, ou 111. Ce qui signifie qu'on pour un cRiceParam donné, on aura $2^{cRiceParam}$ combinaisons possibles. Ce qui signifie également qu'on peut substituer une combinaison par une autre lors de son chiffrement. De même, on peut également chiffrer aussi les signes de QTCs car on peut remplacer les signes positives par ceux négatives et vise versa

V.5.1 Le choix d'espace de chiffrement

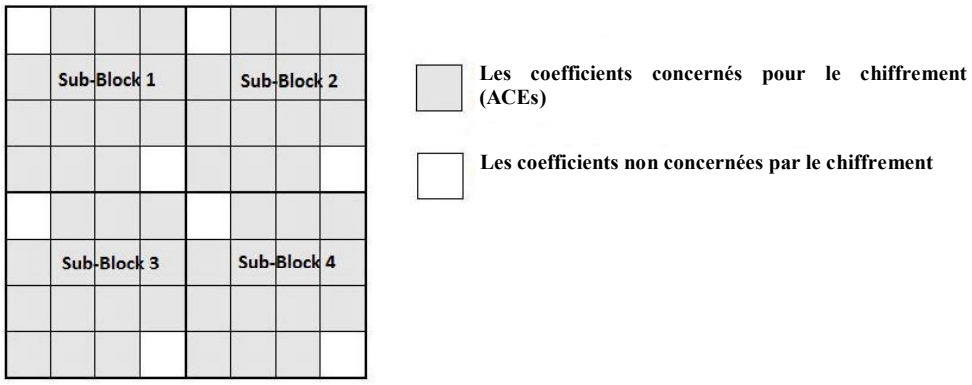


Figure V.3 Exemple d'un bloc de 8×8 QTCs.

Pour chaque blocs de 4×4 QTCs, on choisi seulement les suffixes de QTCs dont ils ne font pas les extrémités de chaque parcours diagonal, car on trouvé expérimentalement que leur chiffrement interrompre le processus de décodage. La figure V.3 montre un exemple d'un bloc d'une unité de transformée contenant 8×8 QTCs. Ce bloc est décomposé en sous-bloc de 4×4 QTCs. Chaque sous-bloc contient 14 QTCs concernés pour le chiffrement. Ces QTCs seront notés par la suite ACEs et vont formé le premier espace de chiffrement noté ES1.

Chaque bloc contient au minimum 14 QTCs chiffrables. Dans notre approche, nous allons sélectionner seulement L_{max} QTCs parmi ces 14 QTCs à chiffrer ; L_{max} sera choisi comme étant une clé secrète pour le chiffrement.

Le deuxième espace de chiffrement noté ES2 concerne tous les signes de QTCs non nuls.

V.5.2 Préparation et chiffrement de plaintext

Notre approche utilise deux plaintexts (voir figure V.4) pour le chiffrement, notés respectivement plaintext1 et plaintext2 et seront préparés pour chaque slice de type intra.

Pour chaque bloc de QTCs 4×4, on empile dans plaintext1 seulement les suffixes des codes de Golomb-Rice de QTCs appartenant à ES1. Le deuxième plaintext va contenir tous les signes de tous les QTCs non nul trouvés selon leur parcours.

(a)

(b)

Figure V.4 Les plaintexts utilisés : (a) le plaintext de suffixes, (b) le plaintext de signe de QTCs non nuls.

Chaque plaintext est chiffré par AES en mode CBC. Les bins chiffrés vont remplacer leur bins à crypter. Chaque plaintext est partitionné en (n+1) blocs notés (X_1, X_2,...., X_n, Y). La longueur de chaque X_i est de 128 bits sauf Y qui aura une taille inferieur a 128. Un vecteur $IV \in \{0,1\}^{128}$ est choisi aléatoirement. Le premier bloc chiffré C_1 est généré en appliquant l'AES à $X_1 \oplus IV$ ($\oplus$ dénote l'opération XOR bit à bit) avec une clé K_1 de 128 bits. Chaque C_i, i=2,..n, C_i est calculé quant a lui par l'AES de $C_{i-1} \oplus X_i$ en utilisant la même clé K_1. Le dernier bloc C_{n+1} est un cas particulier et est calculé par $C_{n+1}=Y \oplus AES(K_2,IV)$, avec K_2 est une clé de 128 différente de K_1.

Après, les bins de ciphertext vont remplacer ceux de plaintext avant leur passage au codage arithmétique binaire comme le montre la figure suivante :

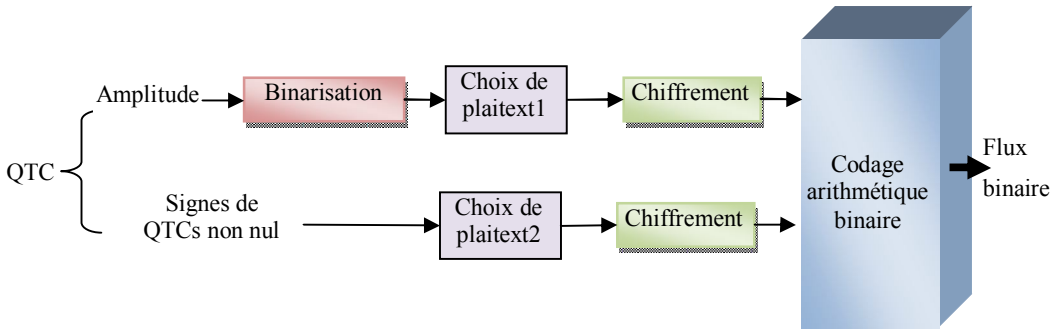


Figure V.5 Les différentes étapes de notre approche.

Le déchiffrement s'est effectué après le décodage arithmétique binaire, en déchiffrant le ciphertext et en replaçant les bins déchiffrés par ceux chiffrés.

Finalement, on rappelle que les clés de notre approche sont K_1, K_2 et L_{max}.

V.6 Résultats expérimentaux

L'approche proposée est implémenté en C++ au moyen de logiciel de référence de HEVC HM ver 10[5]. Le module de chiffrement est inclus conjointent durant la compression comme le montre la figure V.6. De même, le module de déchiffrement est inclus durant le décodage.

Les séquences vidéos de test qu'on a utilisé sont représentées dans le tableau V.1. Aussi, la configuration choisie pour le codage est illustré dans le tableau V.2.

[5] SHM reference software: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/branches/ [dernière visite le 26/06/2015]

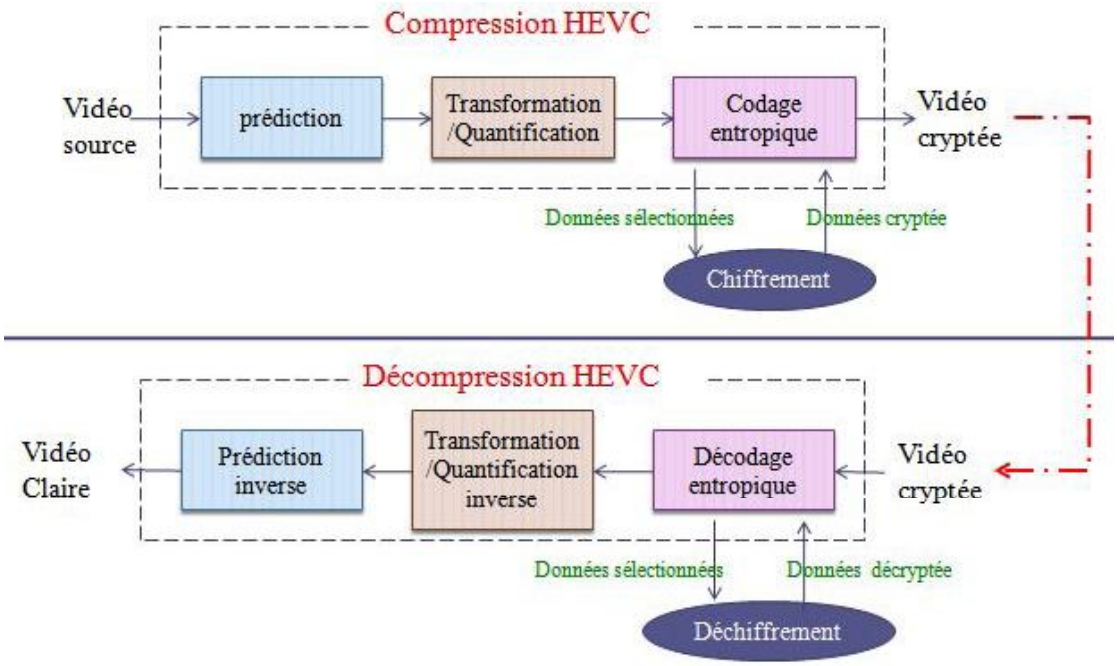


Figure V.6 Application de chiffrement/déchiffrement de durant le processus de codage/décodage de notre approche.

Sequence video	Resolution	Frame-rate
BasketballPass	416×240	50
BasketballDrill	832×480	50
Johnny	1280×720	60
BasketballDrive	1920×1080	50
Traffic	2560×1600	30
YachtRide	3840×2160	120

Tableau V.1 Les différentes séquences de test utilisées.

Parametre	Description	Valeur utilisée
MaxCUWidth	Largeur maximum de CU	64
MaxCUHeight	Longueur maximum de CU	64
MaxPartitionDepth	Pronfondeur maximal	4
IntraPeriod	Periode entre les images Intra	8
GOPSize	La taille de la structure GOP	8
InputBitDepth	8 bit par pixel	8

Tableau V.2 La configuration choisie pour le codage.

V.6.1 Résultats de décodage

Figure V.7 L'image #1 de chaque séquence de test.

Le premier test expérimental s'est effectué en mode low delay où chaque GOP se compose d'une image intra suivie de sept images prédites de type P. La figure V.7 montre la première image de chaque séquence testée. La figure V.8 montre la première image de chaque séquence cryptée pour L_{max}=14 et QP=18. Il est clair que la valeur visuelle est totalement effacée ce qui permet de dire que notre approche offre une confidentialité visuelle suffisante.

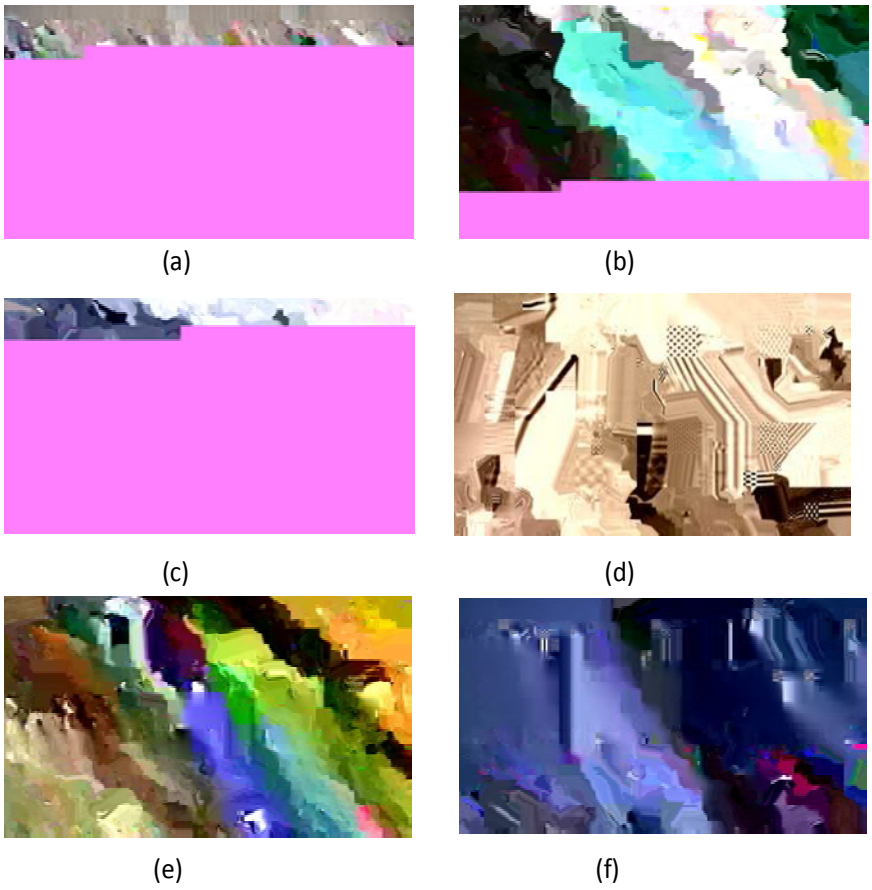


Figure V.8 La première image décodée de chaque flux vidéo crypté.

Dans une deuxième expérience (voir figure V.9), nous avons testé la propagation de chiffrement d'images intra sur les autres images. Nous avons utilisé la séquence BasketballDrill qui contient beaucoup de changement en mouvement. La neuvième image de la séquence est une image intra. Il est clair que son chiffrement s'est propagé vers les images suivantes.

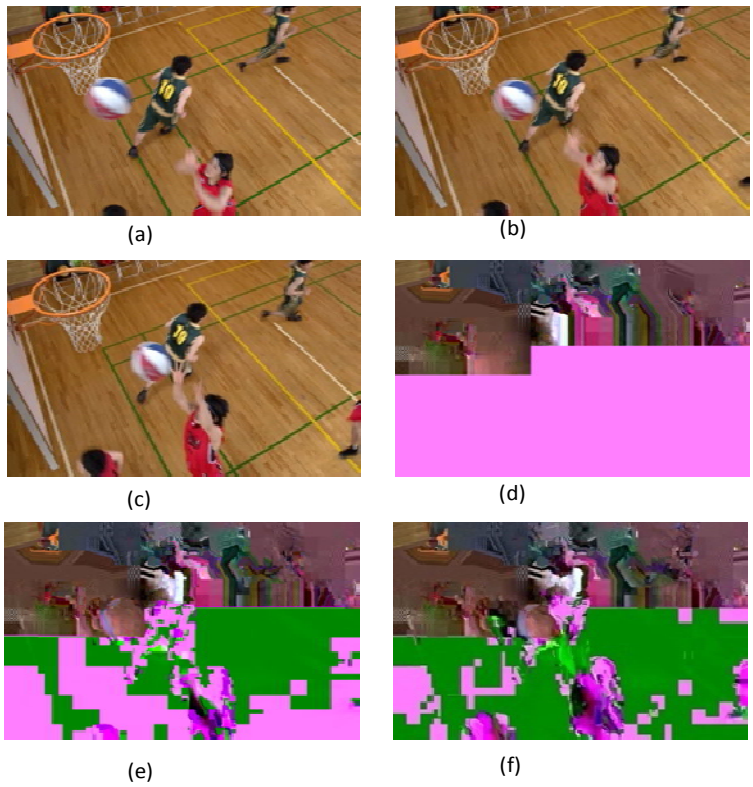


Figure V.9 Propagation de chiffrement d'une image intra vers les autres images de la séquence décodée BasketballDrill avec QP=24 et Lmax=14, (a), (b) et (c) représentent les images #9,#11, et #14 respectivement, et (d), (e), et (f) les images chiffrées correspondantes.

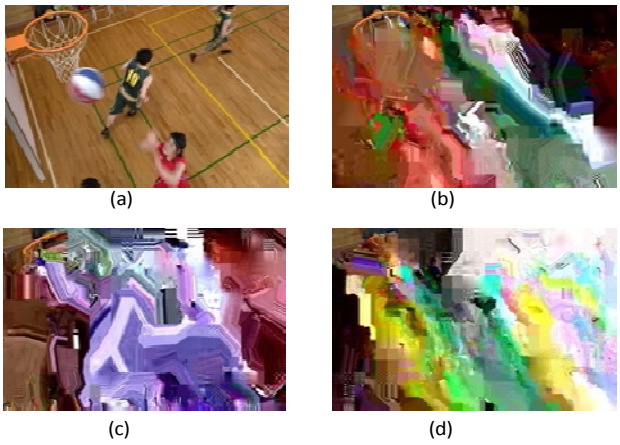


Figure V.10 L'influence de parametre Lmax sur le contenu visuel (QP=24) : (a) l'image originale, (b) resultat de chiffrement pour (b) Lmax=4, (b) Lmax=8, et (c) Lmax=12.

Le but d'une troisième expérience est de tester l'influence de changement de paramètre Lmax sur la qualité visuelle de la sequence cryptée. Où nous avons chiffré la première image de la séquence BasketballDrill en utilisant différentes valeurs de

Lmax. La figure V.10 montre les résultats pour les valeurs de Lmax : 4, 8, et 12 respectivement. Nous constatons qu'à chaque fois on augmente le nombre de QTCs a chiffrer, nous aurons plus de sécurité visuelle. Nous avons trouvé que pour Lmax=4, la dégradation visuelle commence à se dévoiler.

La couleur rose n'est pas un résultat d'une erreur lors de décodage ; elle est juste le résultat quand les valeurs de luminances atteignent 255, et les valeurs de chrominances atteignaient zéro.

V.6.2 Evaluation de l'espace de chiffrement

Sequences	Low delay (%)	Random Access (%)
BasketballPass	9.90	9.06
BasketballDrill	6.90	6.42
Johnny	7.81	6.43
BasketballDrive	3,41	2,28
Traffic	4,01	3,58
YachtRide	5,05	4,78

Tableau V.3 L'espace de chiffrement de toutes les séquences en mode low delay et en mode random access.

L'espace de chiffrement ES (encryption space) est le pourcentage de nombre de bits modifiés après le chiffrement divisé par le nombre total de bits. Le tableau V.3 montre les pourcentages trouvés pour le chiffrement de dix images de chaque séquence en mode low delay et en mode random access avec un QP=18 et Lmax=14. Il est clair que 10% de bits permettent de dégrader considérablement le contenu visuel de chaque séquence.

La valeur de QP	Espace de chiffrement (%)
18	9,90
20	9,12
24	8,22
30	4,10

Tableau V.4 L'influence des valeurs de QP sur l'espace de chiffrement.

Nous avons testé aussi l'impact de changement de pas de quantification QP sur l'espace de chiffrement. Comme le montre le tableau V.4, les valeurs d'espace de chiffrement décroissent de façon inversement proportionnelle avec les valeurs de QP.

Par contre, Les valeurs d'espace de chiffrement décroit de façon proportionnelle avec les valeurs de Lmax comme le montre le tableau V.5.

L_{max}	Percentage d'espace de chiffrement (%)
4	2,14
6	4,12
8	6,16
12	8,01

Tableau V.5 Variation des valeurs d'espace de chiffrement par rapport aux Lmax.

Sequences	Golomb–Rice code (%)	Exp-Golomb code (%)
BasketballPass	87,84	12,15
BasketballDrill	87,42	12,57
Johnny	87,12	12,87
BasketballDrive	87,41	12,58
Traffic	87,36	12,63
YachtRide	87,05	12,94

Tableau V.6 La fréquence d'utilisation des codes de Golomb-Rice et Exp-Golomb par le codeur HEVC.

Finalement, nous avons calculé le pourcentage d'utilisation de chaque code par HEVC. Nous avons trouvé que le code Golomb-Rice est utilisé huit fois beaucoup plus fréquent que celui de Exp-Golomb, ce qui montre le poids de code de Golomb-Rice et son impact pour la sécurité visuelle.

V.6.3 Evaluation de qualité visuelle

Nous avons testé la sécurité visuelle au moyen de deux mesures PSNR et SSIM [80]. Le tableau V.7 donne les valeurs de PSNR calculées entre la première image originale et l'image cryptée en mode low delay avec QP=18 et Lmax=14. Les valeurs minimales trouvées pour les séquences cryptées sont ceux de luminance. Ces basses valeurs signifient qu'une très haute dégradation visuelle est atteinte.

SSIM (Structural SIMilarity) est une mesure de similarité de structures entre deux images numériques, et est considéré comme une mesure subjective. Dans une autre expérience, nous avons testé l'influence de Lmax et de QP sur le contenu visuel de video chiffrée en mesurant à chaque fois les valeurs de PSNR et de SSIM

correspondantes (Tableau V.8). Les valeurs de PSNR relatives à la composante de luminance Y sont tous au dessous de 16, et les valeurs de SSIM sont tous au dessous de 0.4, ce qui signifie que notre approche offre un niveau sécurité satisfaisant selon les conditions données par [81].

	PSNR Y (dB)		PSNR U (dB)		PSNR V(dB)	
Sequence	Orig.	Enc.	Orig.	Enc.	Orig.	Enc.
BasketballPass	45.77	8.64	47.23	26.05	47.46	22.21
BasketballDrill	45.19	11.9	45.99	16.80	47.10	18.47
Johnny	46.26	9.33	49.44	21.09	49.87	23.29
BasketballDrive	46.45	9.93	46.35	9.21	48.16	13.76
Traffic	46.25	6.89	45.52	14.66	47.04	15.45
YachtRide	47.59	8.80	48.92	12.11	48.04	10.27

Tableau V.7 Les valeurs de PSNR trouvées pour toutes les séquences vidéo utilisées.

Evaluation selon L_{max}			Evaluation selon QP		
Valeur de L_{max}	PSNR Y (dB)	SSIM	Valeur de QP	PSNR Y (dB)	SSIM
3	15.65	0.264	16	10.89	0.222
4	12.33	0.287	18	8.64	0.020
5	13.03	0.286	22	9.19	0.164
10	11.77	0.148	26	11.31	0.130
14	8.64	0.020	28	10.12	0.110

Tableau V.8 L'impact de changement de Lmax et de QP sur les valeurs de PSNR et de SSIM pour la sequence BasketballPas.

V.6.4 Evaluation de performances de l'approche proposée

Les tests expérimentaux se sont performées dans une architecture matérielle disposée d'un processeur Intel 2.3GHz Dual-Core T4500 avec 3 GB de RAM. Le tableau V.9 nous donne le temps d'exécution de processus de codage avec ou sans chiffrement exprimés en millisecondes (Ms) d'une image de chaque séquence de test. La différence trouvée entre le temps de codage sans et avec chiffrement est négligeable, et elle nous donne le temps nécessaire pour le chiffrement et le remplacement des bins cryptés avant le codage arithmétique binaire.

Sequences	Temps de codage (Ms)	
	Sans chiffrement	Avec chiffrement
BasketballPass	22.94	23.08
BasketballDrill	90.81	90.90
Johnny	162.62	162.69
BasketballDrive	444.15	444.23
Traffic	845.12	845.94
YachtRide	1520.10	1521.01

Tableau V.9 Le temps de codage de la première image sans/avec chiffrement.

V.6.5 Espace de clé

Les deux clés K_1 et K_2 sont sur 128 bits, et la clé Lmax est sur 4 bits (car elle représente 14 combinaisons possibles). Donc l'espace de clé est égal a $2^{128}\times2^{128}\times2^{4}$ clé possible. On constate donc que notre approche est résistante en termes de sécurité calculatoire.

V.7 Conclusion

Dans le présent chapitre, nous avons présenté notre approche qui permet de chiffrer sélectivement les codes de Golomb-rice et les signes de QTCs au moyen de AES en mode CBC. Après avoir expliqué les différents éléments syntaxiques utilisés pour le codage de QTCs, nous avons exposé la problématique où nous avons vu que les travaux antérieurs n'ont pas donnée des solutions satisfaisantes pour chiffrer les amplitudes de QTCs. Après, nous avons entamé notre approche qui permet de chiffrer les QTCs selon le dernier codage entropique standardisé de HEVC. Le décodage de flux binaire crypté est une preuve satisfaisante de conformité de la notre. La haute dégradation visuelle est une autre preuve sur la sécurité visuelle qu'elle offre notre approche, où elle est justifiée à l'aide de mesure de PSNR et SSIM. Aussi, nous avons vu que seulement une 10% de flux binaire comprimé peut dégrader suffisamment le contenu visuel de la vidéo et d'avoir un niveau satisfaisant de sécurité visuelle.

Le document standard de HEVC donne seulement le format de flux binaire et l'algorithme de son décodage. Cependant, il ne détaille pas le codage des éléments syntaxique qui est très important pour concevoir une approche de chiffrement selon des bases théoriques consistantes. C'est pour cette raison qu'on a localisé

expérimentalement les codes chiffrables qui permettent d'assurer la conformité et le décodage de flux binaire crypté.

Dans le prochain chapitre, nous allons plus approfondir dans le codage de QTCs en proposant dans une deuxième approche, une solution améliorée de la première approche qui sera plus conforme et plus robuste avec des résultats remarquables en termes de sécurité visuelle.

Chapitre VI.

Un nouveau schéma de chiffrement sélectif conforme pour la sécurité de HEVC/H.265

VI.1 Introduction

Après sa standardisation en Avril 2013, et la publication de son premier document standardisé, la norme de codage vidéo HEVC va certainement dominer tous les champs de recherche multimédia, surtout ceux concernant la sécurité de flux vidéo compressé.

HEVC est attendue et prévue pour être le codec de l'ère Ultra HD. Elle a été développée pour répondre à une large gamme de besoins de forte compression d'images animées à haute définition pour diverses applications telles que la visioconférence, la sauvegarde, la diffusion télévisuelle, le trafic Internet et la communication. Elle est aussi conçue pour permettre une utilisation souple de la représentation vidéo codée dans une large gamme d'environnements de réseaux.

CABAC est le seul codeur entropique adopté depuis HEVC, où il est constitué de trois étapes principales : la binarisation, le choix de tables de contextes, et le codage arithmétique binaire.

Avant la standardisation de HEVC, le codage entropique des données fréquentielles a occupé un champ de recherche très vaste. Il a passé par plusieurs évolutions : l'annulation de CAVLC et l'adoption de CABAC depuis WD ver 6 [75], la publication de premier codage des données transformées en Décembre 2012 [82], la publication de deuxième approche de codage des données fréquentielles en 2013 [83], et la publication de la version finale de l'approche utilisée pour la binarisation de QTCs en décembre 2014 [84].

Z. Shahid et al. ont proposé dans leurs travaux [73] et [74], des approches pour le chiffrement sélectif de HEVC par la protection de QTCs (Quantized transform coefficients) au moyen de l'AES, où ils ont adopté le document WD ver 6 pour surmonter le défi de chiffrement de QTCs durant le module de CABAC. Le codage de QTCs employé dans WD ver 6 est publié dans l'article [82]. Malheureusement, Ce codage est changé entièrement, et est remplacé par un autre codage publié en [83] dont le procédé de binarisation est publié dans [84]. En conséquence, les approches de Z. Shahid et al. [73][74] sont devenues non appropriées et nécessitent une mise à niveau par la conception d'autres approches alternatives car le format de flux binaire de HEVC est carrément changé dans la partie qui représente les données fréquentielles.

Dans ce chapitre, nous allons proposer une nouvelle approche de chiffrement sélectif [86] s'appliquant durant le module de codage entropique de HEVC, où nous allons protéger les données fréquentielles (amplitudes et signes de QTCs) selon le dernier codage entropique de QTCs vu dans [84]. Premièrement, nous allons commencer par donner une brève description de codage entropique de QTCs selon [84]. Ensuite, nous allons exposer la problématique par mentionner les différences qui existent entre les codages entropiques antérieurs et présents de QTCs. Aprés, nous allons expliquer notre approche de chiffrement sélectif. Notre approche sera soumise ultérieurement à une évaluation par des tests expérimentaux. Finalement, on clôturera le chapitre par une conclusion.

VI.2 Une brève description de codage entropique de QTCs

Après leur transformation et leur quantification, les blocs de l'erreur résiduelle quantifiée ayant une taille plus de 4×4 QTCs sont divisés en sous-blocs de 4×4 QTCs. Chaque sous blocs contient 16 QTCs qui seront balayés selon un parcours diagonal inverse en zigzag, en commençant d'abord par les hautes fréquences, en passant par les basses fréquences, et en terminant finalement par le DC.

Le codage de QTCs selon l'article correctif [84] est assuré par un algorithme qui utilise cinq éléments syntaxiques qui sont : *significant_coeff_flag*, *coeff_abs_level_greater1_flag*, *coeff_abs_level_greater2_flag*, *coeff_sign_flag*, et *coeff_abs_level_remaining*. Les trois premiers éléments syntaxiques sont codés mode

regulier de CABAC, alors que les deux éléments syntaxiques restants sont codés en mode bypass qui initialise le codage arithmétique binaire par assumer 0.5 de probabilité pour chaque décision binaire qui s'appelle bins lors de codage de chaque chaine binaire.

coeff_abs_level_remaining est le seul éléments syntaxique qui nécessite la binarisation pour son codage comme le montre la figure 6.1, où deux codes sont employé pour atteindre ce but qui sont : le code de Exp-Golomb d'ordre p EG_p et le code de Golom-Rice d'ordre p TR_p (pour Truncated Rice code).

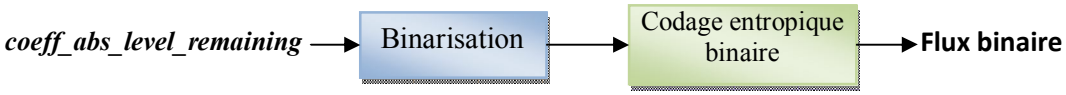


Figure VI.1 Le codage de l'élément syntaxique coeff_abs_level_remaining.

Pour une valeur entière donnée x, et pour un paramètre d'ordre p donné :

- le code de TR_p s'obtient par la concaténation de préfixe q et de suffixe k. Le préfixe est le code unaire de $q = round(\frac{x}{2^p})$. Le suffixe k est la représentation binaire de $x - q \times 2^p$ avec un nombre de bits egal a p.
- Le code EG_p est juste la concaténation de préfixe de suffixe. Le préfixe est le code unaire de $\ell(x) = \log_2(\frac{x}{2^p} + 1)$, alors que le suffixe est calculé avec $x+2^p(1-2^{l(x)})$.

La valeur de *coeff_abs_level_remaining* est calculée par :

$$coeff\_abs\_level\_remaining = |QTC| - baselevel \quad \text{(VI.1)}$$

avec baselevel est un paramètre qui prend la valeur 1, 2, ou 3, et est calculé en fonction des QTCs précédemment codés [83].

Pour un paramètre d'ordre p, *coeff_abs_level_remaining* est binarisé au moyen de code de TR_p si sa valeur est inferieur a $TR_{max}[p]$, Autrement, son code est obtenu par la binarisation de la partie (coeff_abs_level_remaining - $TR_{max}[p]$) a l'aide EG_{p+1} préfixé par la chaine ''1111'' comme le montre la figure VI.2.

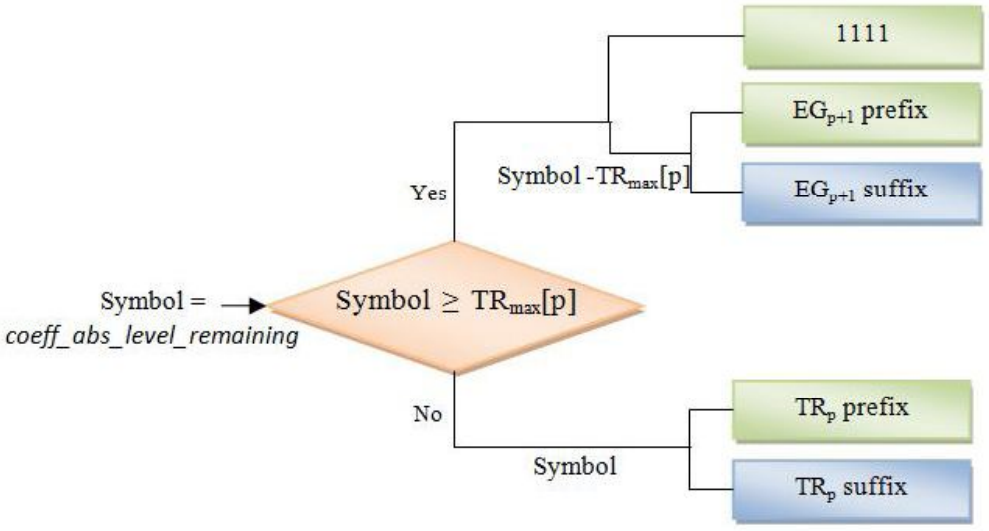


Figure VI.2 Le schéma de binarisation de coeff_abs_level_remaining selon [84].

Les valeurs de seuil $TR_{max}[p]$ dépendent étroitement de la valeur de paramètre d'ordre p de la manière suivante :

$$TR_{max}[p]=4\times 2^{p} \quad ,p \in \{0,1,2,3,4\} \ . \tag{VI.2}$$

L'ordre de paramètre p est initialisé à zero pour chaque bloc de l'erreur résiduelle. Apres l'encodage de chaque QTCs, il sera mis à jour suivant la relation :

$$p_{next} \leftarrow min\,(p+1,4), si\ |QTC|>3\times 2^{p} \tag{VI.3}$$

Dans le tableau VI.1, on trouve tous les codes qui permet de représenter *coeff_abs_level_remaining* quand $p = 2$. L tableau a gauche représente les codes TR_2, alors le tableau a droite représente les codes EG_3.

VI.3 Problématique

La binarisation de *coeff_abs_level_remaining* a connu beaucoup de changement selon le tableau VI.2. Premièrement, Les valeurs de seuil $TR_{max}[p]$ ont été totalement modifiées, où elles prennent des valeurs multiples de quatre. EG_0 est substitué par EG_{p+1}. Finalement, la règle de mise à jour de paramètre de l'ordre est aussi modifiée.

symbole	préfixe	suffixe
0	0	00
1	0	01
2	0	10
3	0	11
4	10	00
5	10	01
6	10	10
7	10	11
8	110	00
9	110	01
10	110	10
11	110	11
12	1110	00
13	1110	01
14	1110	10
15	1110	11

symbole	préfixe	suffixe
16	11110	000
17	11110	001
18	11110	010
19	11110	011
20	11110	100
21	11110	101
22	11110	110
23	11110	111
24	111110	0000
25	111110	0001
26	111110	0010
27	111110	0011
28	111110	0100
29	111110	0101
30	111110	0110
31	111110	0111

Tableau VI.1 Les codes utilisés pour représenter les valeurs de coeff_abs_level_remaining quand p=2.

	WD6	[84]
Valeur codée	$\|QTC-3\|$	$\|QTC\| - baselevel$
Les valeurs de p	{0,1,2,3,4}	{0,1,2,3,4}
$TR_{max}[p]$	$\{7,14,26,46,78\}$	{4,8,16,32,64}
valeur< $TR_{max}[p]$	TR_p code	TR_p code
valeur≥$TR_{max}[p]$	Code Exp-Golomb d'ordre zero	Code de Exp-Golomb d'ordre (p+1)
La règle de mise a jour de p	$p_{next}[\|QTC\|] = \{3,5,12,24,\infty\}$	$p_{next} \leftarrow min\ (p+1,4)$,if $\|QTC\|>3\times2^p$

Tableau VI.2 Les différences qui existent le codage de coeff_abs_level_remaining selon WD6 et selon [84].

Ces changements profonds sont des preuves satisfaisantes que les travaux antérieurs de Z. Shahid et al. [73][74] deviennent non applicables pour la protection de vidéos HEVC. Aussi, dans [73], Ils ont proposé de diviser les codes TR_p en sous codes TR_p. Cette division signifie que lors de décodage, le décodeur va trouver des codes, c'est-à-dire des QTCs additionnels sans signes, ce qui va interrompre le processus de décodage en revanche.

VI.4 L'approche proposée

A la lumière de [84], la conception d'un système de chiffrement sélectif basée sur la protection de *coeff_abs_level_remaining* doit tenir en compte les considérations suivantes :

1) On peut chiffrer seulement les suffixes de TR_p et ceux de EG_{p+1}.

2) L'ordre p_{next} ne devrait pas se changer après le chiffrement, afin que chaque code doive garder son propre ordre.
3) Le format des codes (TR_p, EG_{p+1}) devrait être préservé.

Autrement dit, si on suppose qu'un bloc contient six QTCs non nuls dont leurs amplitudes sont codées comme suit : TR_0, TR_1, TR_2, TR_2, TR_3, et EG_1, alors le bloc chiffré correspondant devrait contenir aussi six QTCs dont leurs amplitudes devront être codées comme ceux de bloc en clair, c'est-à-dire il faudrait que le décodeur devrait trouver la séquence suivante après le décodage de bloc chiffré: TR_0, TR_1, TR_2, TR_2, TR_3, et EG_1.

Afin de répondre à ces considérations, on devrait faire une petite analyse sur le comportement probable des codes après le chiffrement. Dans le tableau 6.1, les codes TR_p qui gardent leurs ordres après le chiffrement sont seulement ceux de 0 jusqu'à 11. Où la modification de chaque suffixe de TR_p donne un autre code TR_p inferieur à $TR_{max}[2]=12$. Par contre, la modification des codes TR_p colorés en gris vont changer l'ordre p, car si on modifie le suffixe de code 12, on va trouver soit 13, 14, ou 15 qui vont changer l'ordre p_{next} à $(p+1)$ après ses modification. Supposons que QTCe est le coefficient chiffré correspondant à QTC, et que l'amplitude de QTC ($|QTC| - baselevel$) est binarisé avec le code TR_p qui est égal en valeur décimale à $prefix \times 2^p + suffix$. Après le chiffrement de suffixe de QTC, la valeur de QTCe sera égal à$prefix \times 2^p + suffix_e + baselevel$. Cette valeur devrait être inferieur a $3 \times 2^p - 1$pour que l'ordre p_{next} ne changerait pas après le chiffrement.

Aussi afin de garantir plus de conformité de codage, il faut chiffrer seulement les codes TR_p vérifiant la condition suivante donnée par l'expression logique $prefix \times 2^p + 2^p - 1 + baselevel \leq 3 \times 2^p - 1$.

Le chiffrement de suffixes des codes EG_{p+1} ne casse pas la règle (6.3), car tous les codes EG_{p+1} sont supérieurs à $TR_{max}[p]$ sans et avec chiffrement.

Dans notre approche, nous allons préparer deux plaintexts, le premier contient les suffixes de TR_p vérifiant la condition suivante $prefix \times 2^p + 2^p - 1 - baselevel \leq 3 \times 2^p - 1$ et les suffixes de EG_{p+1}. Le deuxième plaintext contient tous les signes de QTCs non nuls. HEVC regroupe tous les signes de chaque bloc de 4×4 QTCs et les encode ensemble. Ce qui nous facilite la construction de ce deuxième plaintext.

Après leur construction, les plaintexts seront chiffrées comme est expliqué dans la sous-section V.5.2. La figure VI.3 résume les différentes étapes de chiffrement proposées. Le processus de chiffrement doit se faire avant l'étape de codage arithmétique binaire BAC (binary arithmetic coding)

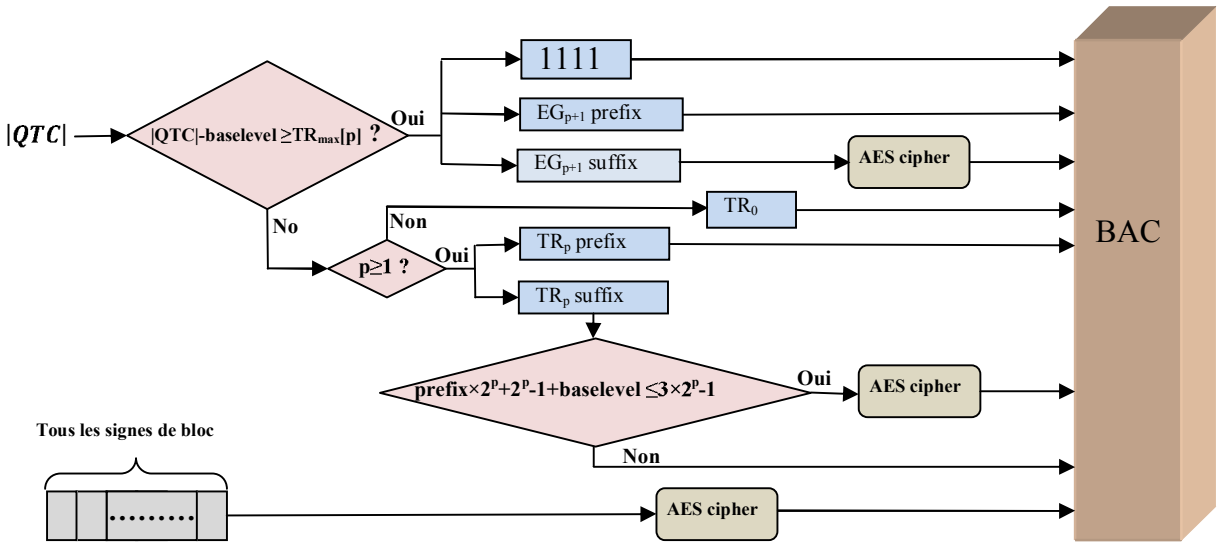


Figure VI.3 Un schéma fonctionnel qui explique les différentes étapes de l'approche proposée.

VI.5 Résultats expérimentaux

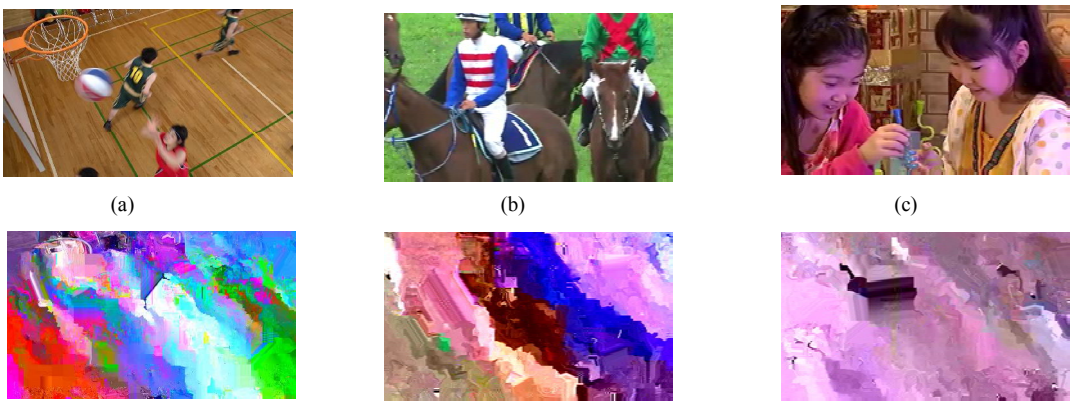
Classe	Résolution	Frames/seconde	Vidéo	Images à encoder
A	2560x1600	30	Traffic(S01), PeopleOnStreet(S02), Kimono,	100
B	1920x1080	24	Kimono1 (S03), ParkScene (S04)	100
C	1920x1080	50-60	Cactus (S05), BDrive (S06),BQTerrace (S07)	100
D	832x480	30-60	BasketballDrill (S08), BQMall (S09), PartyScene (S10), RaceHorses (S11)	100
E	416x240	30-60	BPass (S12), BQSquare (S13), BlowingBubbles (S14), RaceHorses (S15)	100
F	1920x1080	60	Vidyo1 (S16), Vidyo2 (S17), Vidyo3 (S18)	100

Tableau VI.3 Les 18 séquences de test utilisées pour la validation de notre approche.

Comme notre approche est un système de crypto-compression appliqué durant le module de codage entropique, nous avons inclus notre cryptosystème dans le processus de compression. De même, nous avons inclus le processus de déchiffrement durant la phase de décodage. Pour cela, nous avons implémenté notre approche en langage C++ par l'utilisation de logiciel de référence de HEVC ver 10.00. Les séquences de tests utilisés (voir le tableau VI.3) répondent aux conditions de test

communes spécifiées dans [85]. Ces séquences varient selon plusieurs caractéristiques comme la texture, le mouvement, les objets,…

VI.5.1 Evaluation de la qualité visuelle de résultats de décodage



(a) (b) (c)

(d) (e) (f)

Figure VI.4 Résultats visuels de notre approche décodés pour QP=18 en mode Intra only : (a), (b), et (c) sont images intra originales, (d), (e), et (f) représentent les images chiffrées correspondantes.

Premièrement, nous avons testé notre approche par l'encodage de 100 images de chaque séquence en mode Intra only à un pas de quantification $QP = 18$. La figure VI.4 montre les résultats obtenus pour trois séquences vidéo. Il est clair que dans les images cryptées décodées, une haute dégradation visuelle est atteinte, et la valeur commerciale de chaque image est totalement effacée.

(a) (b) (c)

Figure VI.5 Résultats de notre approche décodés à QP=28 : (a) l'image originale, (b) image P encodée en mode low delay (c) image de type B encodée en mode random access.

Deuxièmement, nous avons testé le résultat de notre approche en mode low delay et en mode random access. La figure VI.5 montre le résultat de chiffrement de la séquence à un pas $QP = 28$ encodé (b) en mode low delay, et (c) en mode random access. On remarque que notre approche est appropriée aussi dans ces modes avec

une confidentialité visuelle satisfaisante. Le mode low delay est préféré pour des applications de communication alors que le mode random access est favorisé pour les applications destinées à la sauvegarde.

	Mode Intra only			Mode Low delay			Mode Random access		
	Y	Cb	Cr	Y	Cb	Cr	Y	Cb	Cr
S01	9.3296	15.0736	11.5338	8.5185	14.8263	12.4571	9.3222	12.4174	12.9084
S02	9.3530	15.5290	14.2688	10.3549	16.7462	16.5045	8.0752	11.7164	11.7153
S03	10.0783	11.6349	13.4709	10.2610	12.4442	12.6144	9.7153	14.6438	16.6390
S04	9.9165	15.5669	14.8953	7.6580	15.0567	15.5930	8.2285	13.7008	14.8733
S05	9.2302	14.1618	14.5752	9.7136	12.7344	12.1320	10.2523	13.7524	13.8768
S06	8.0488	11.4877	15.5186	8.3074	15.0308	12.7250	8.3072	14.9717	14.8359
S07	8.2198	12.3925	15.4546	8.7685	15.1708	11.5467	9.7965	15.6217	14.2683
S08	10.1595	13.7344	12.6708	9.1436	11.4080	14.4573	8.0660	13.1013	14.8839
S09	7.5860	15.7875	14.4937	10.3282	12.5287	15.1002	8.3625	14.9721	14.2633
S10	8.9697	15.8244	11.9890	8.7532	12.3442	14.2796	7.7733	13.4970	15.3263
S11	8.0038	11.7881	11.1527	10.4492	15.0070	13.5544	9.2286	16.0516	14.1350
S12	10.4360	15.8530	14.7204	8.4044	16.0664	14.8667	9.5501	15.9975	16.9622
S13	9.6381	15.7858	13.5001	9.6033	13.0668	14.8857	9.1398	12.5386	12.3121
S14	9.0014	13.4269	13.3996	9.4990	15.6831	15.0741	8.7772	14.6808	11.6348
S15	8.9133	15.0014	15.5236	9.1174	15.0520	14.8147	9.4333	14.4935	11.6582
S16	7.6789	11.7094	14.0493	9.5943	11.0403	16.6710	9.4429	14.2444	11.3815
S17	9.5459	13.1088	14.0883	9.4996	14.6130	12.2536	9.5371	16.2196	13.4275
S18	7.6273	15.5787	15.2972	8.0344	13.3206	15.2557	9.4074	16.2196	13.6902
moyenne	**8.9853**	**14.0803**	**13.9223**	**9.2227**	**14.0078**	**14.1548**	**9.0231**	**14.1783**	**13.8218**

Tableau VI.4 PSNR moyen de toutes les séquences de test encodées à un pas QP=18.

	Mode Intra only			Mode Low delay			Mode Random access		
	Y	Cb	Cr	Y	Cb	Cr	Y	Cb	Cr
S01	9.3353	17.4925	16.8796	9.7456	18.2675	18.6889	9.5724	17.5338	22.0385
S02	9.5315	16.6851	19.2407	9.1159	18.9106	19.9781	10.6049	20.1064	18.6508
S03	11.2374	17.5660	19.2479	10.2636	17.3849	20.2468	10.2112	17.3031	18.4774
S04	9.1786	16.5475	19.9501	10.9036	16.6243	18.4712	9.3796	21.2654	23.2280
S05	9.6062	19.9787	18.6209	10.5777	18.1401	19.2777	10.9548	19.4114	23.7490
S06	9.1344	16.4243	16.3181	9.2247	17.6311	18.5954	9.2515	19.3014	21.4338
S07	10.1043	20.1852	20.1524	9.2022	19.5465	16.1164	9.7352	16.9493	18.1706
S08	9.0332	17.7956	19.6017	10.9431	19.5113	18.0094	9.5934	16.4969	16.7537
S09	11.2430	16.2133	17.2868	11.2628	19.0083	18.9084	10.3272	17.4039	23.1252
S10	9.4916	17.5407	18.4465	10.3344	16.6008	18.3454	9.2287	20.7221	20.9701
S11	9.2334	19.3118	20.4315	9.2729	16.0970	17.6754	10.0133	21.0109	24.0589
S12	9.7684	19.5761	19.2206	11.0645	18.5193	20.2171	9.2621	19.8480	16.5187
S13	10.1401	18.4521	19.7754	9.8452	17.3537	19.7329	9.2807	19.9885	20.9695
S14	9.2542	19.0880	17.9497	9.7349	20.2273	19.8209	10.9611	17.2644	18.4234
S15	11.4885	20.0213	18.1178	10.8658	20.4141	17.6764	9.7289	19.1683	23.0357
S16	9.8302	16.2466	18.5232	9.0258	17.2898	18.6693	10.5088	20.4585	17.6234
S17	9.7434	17.3665	17.2109	9.1211	19.6037	19.9265	11.4111	18.2211	19.7615
S18	9.1551	16.2079	19.3706	10.6698	20.0325	20.2008	10.0812	21.4364	19.3440
moyenne	**9.8060**	**17.9277**	**18.6855**	**10.0652**	**18.3979**	**18.9198**	**10.0059**	**19.1050**	**20.3518**

Tableau VI.5 PSNR moyen de toutes les séquences de test encodées à un pas QP=32.

Cette confidentialité visuelle est évaluée au moyen de deux métriques PSNR et SSIM. Le tableau 6.4 et 6.5 nous donnent les valeurs moyennes de PSNR calculées pour toutes les séquences à un pas QP=18 et QP=32 respectivement. Les basses

valeurs obtenues dans tous les modes dans toutes les composantes de couleur (luminance, et chrominances) qui sont tous inferieures à 25 dB et surtout les basses valeurs obtenues pour la luminance, montrent qu'un haut niveau de la sécurité visuelle est atteint.

QP	PSNR Y(dB)		SSIM	Bits modifies après le chiffrement	ES(%)
	Originale	cryptée			
12	52.45	8.02	0.019	232011	21.12
18	48.75	8.75	0.12	198255	17.64
24	42.11	9.15	0.26	117525	18.15
30	37.70	10.96	0.20	80657	15.75
36	33.95	10.85	0.32	52336	13.96
42	27.15	12.63	0.28	21045	13.12

Tableau VI.6 L'influence de changement de QP sur la qualité visuelle de la séquence cryptée PARTYSCENE encodée en mode low delay.

Nous avons testé aussi l'impact de variation de QP sur la qualité visuelle d'images décodées, où nous avons calculé à chaque fois les valeurs de PSNR et de SSIM associées. Pour cela, nous avons chiffré la première de la séquence PARTYSCENE en mode low delay à des valeurs équidistantes de QP. Le tableau VI.6 montre les valeurs moyennes trouvées, et aussi le nombre de bits affectés par le chiffrement dans chaque test et l'espace de chiffrement. Les basses valeurs obtenues de SSIM et de PSNR dans tous les niveaux de compression constituent une grande preuve de la sécurité visuelle offerte par notre approche. Aussi, il est remarqué que des petits pourcentages de données binaires chiffrées ont été trouvés, ce qui signifie que les données concernées par le chiffrement permettent de contrôler la qualité visuelle de séquences cryptées. En résumé, notre approche est applicable dans tous les niveaux de compression où les données choisies pour le chiffrement sélectif sont toujours présentes dans les bas niveaux comme dans les hauts niveaux de compression.

Finalement, nous avons calculé les pourcentages de chiffrement appliqué dans tous les modes avec un pas QP=18 dont les résultats sont donnés dans le tableau VI.7. Nous avons trouvé des valeurs maximales dans le mode Intra only car le chiffrement est appliqué pour chaque image. Aussi nous avons trouvé des pourcentages significatives dans les autres modes ce qui signifie la fiabilité de notre approche dans tous les modes de HEVC.

Sequence	Intra only	Low delay	Random access
S01	30.6849	17.8418	19.1556
S02	30.6581	20.0715	16.8726
S03	31.7781	22.6669	16.9228
S04	36.4829	20.9676	19.5627
S05	33.0044	18.2084	21.4779
S06	32.4337	21.2471	17.9543
S07	30.5235	23.1099	18.9205
S08	37.2359	19.6139	20.4343
S09	32.4634	19.5909	22.7974
S10	36.1781	17.6444	16.3734
S11	34.2303	15.6055	21.0829
S12	32.2012	16.5468	19.9108
S13	36.4560	15.7904	18.6830
S14	35.6712	18.9397	22.5161
S15	32.9521	15.0793	18.3409
S16	35.9188	22.7777	16.5213
S17	36.6368	20.4633	20.3833
S18	32.6485	15.0121	20.3040

Tableau VI.7 Les pourcentages d'espace de chiffrement de toutes les séquences encodées à un pas QP=18.

VI.5.2 Performance de l'approche proposée

	Temps d'encodage (sec)			Temps de décodage (sec)		
Nombre d'images encodées	**Sans SE**	**Avec SE**	**Différence**	**Sans SE**	**Avec SE**	**Différence**
10	4930.25	4931.15	0.90	58.15	58.35	0.20
30	14850.36	14851.60	1.24	168.23	168.91	0.68
50	24550.91	24553.06	2.15	280.37	282.22	1.85
100	48609.12	48613.03	3.91	518.24	521.86	3.62

Tableau VI.8 Temps de codage/décodage obtenus par le chiffrement (SE) de la séquence KIMONO en mode low delay avec un pas QP=18.

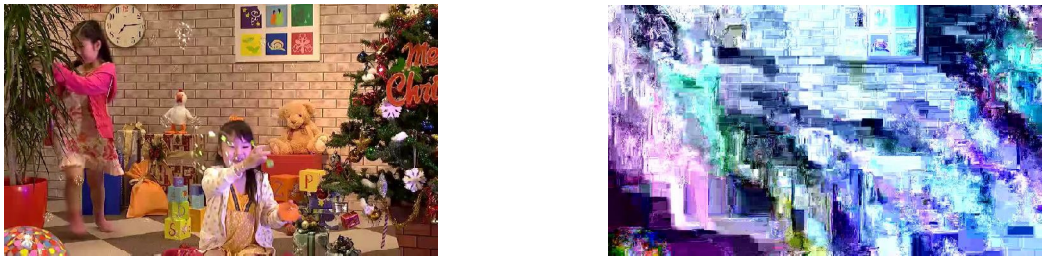
Nous avons performé notre approche dans une configuration matérielle équipée d'un processeur Dual-Core T4500 avec 3GB de RAM. Afin d'évaluer la rapidité de notre approche, nous avons calculé le temps d'exécution pour chaque processus de codage/décodage sans ou avec chiffrement (Tableau VI.8), où nous avons appliqué ce test sur la séquence KIMONO en mode low delay. La différence temporelle qui existe entre l'encodage avec et sans chiffrement est très négligeable, ce qui permet de déployer notre approches dans des applications de communication comme la diffusion

en continu HD (streaming HD), ou dans des applications de vidéoconférence à haute définition.

VI.5.3 Sécurité et robustesse de l'approche proposée

Notre approche utilise seulement deux clés pour le chiffrement, chacun d'entre eux est sur 128 bits. Donc, l'espace de clé contient $2^{128+128} = 2^{256}$ clés possibles. On déduit alors que notre approche est suffisamment robuste sur en terme de sécurité calculatoire.

Dans une deuxième expérience (voir figure VI.6), nous avons testé la sensitivité de notre approche au changement probable de clé. Où nous avons supposé que l'adversaire a pu trouver une partie de clé qui manque seulement une ou deux bits. Nous avons trouvé que notre approche est très sensitive au petit changement de la clé. Dans cette expérience, nous avons chiffré la séquence PartyScene en utilisant la clé $K_1 = FFFFFFFFFFFFFFFF$, et nous avons tenté de déchiffrer le flux compressé à l'aide de la fausse clé $K_1 = FFFFFFFFFFFFFFFE$.

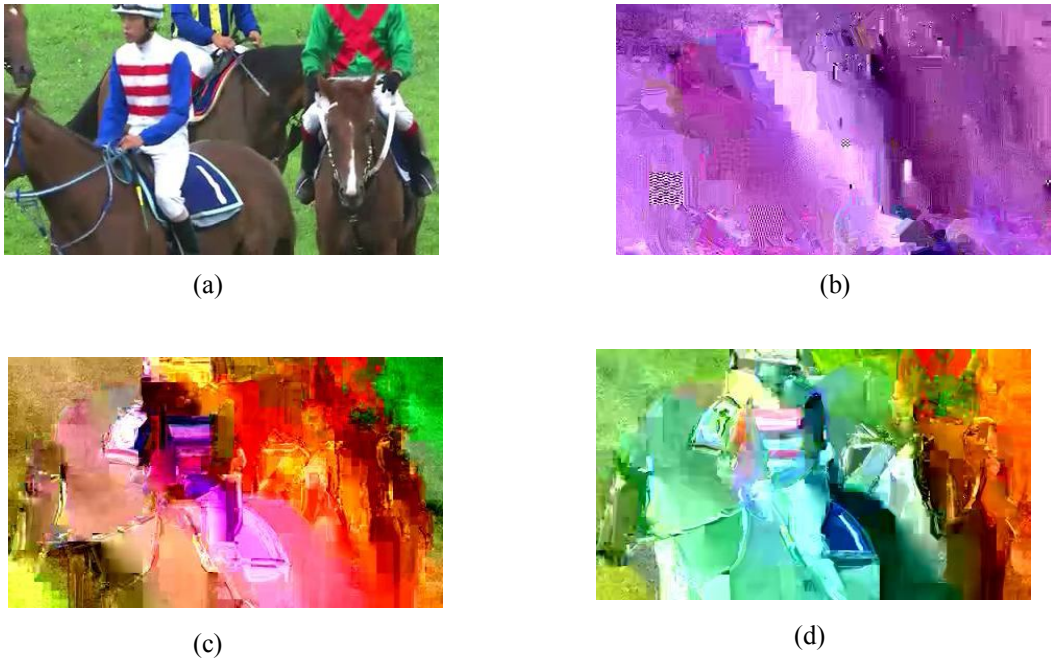


(a) (b)

Figure VI.6 Résultat de décodage en utilisant : (a) la vraie clé, (b) la fausse clé.

Dans le but de simuler l'attaque à texte clair connu, nous avons supposé que l'adversaire tente de rendre le contenu visuel de la vidéo chiffrée visible par remplacer aléatoirement les données chiffrées par d'autres valeurs. La restitution de ces données chiffrées est supposée possible grâce aux techniques d'attaque de force brute. Dans notre cas, nous avons remplacé les données chiffrées de suffixes de codes TR_p/EG_{p+1} par des zeros. La figure VI.7 montre le résultat de cette simulation où elle est appliquée sur la séquence RiceHorse à des niveaux différents de compression. La

qualité visuelle d'images attaquées confirme la robustesse de notre approche sur ce type d'attaque.



(a) (b)

(c) (d)

Figure VI.7 Résultat de simulation d'attaque à texte clair connu appliqué a la séquence RiceHorse : (a) l'image originale, et les autres sont des images chiffrées attaquées quand (b) QP=18, (b) QP=28, (c) QP=34.

VI.5.4 Etude comparative

Nous avons choisi pour notre étude comparative seulement les approches de chiffrement de HEVC appliquées durant le codage entropique qui sont présentés dans le tableau ci-dessous. Où nous avons utilisé plusieurs critères de comparaison

Les travaux de Wallendael et al. [77] s'articulent principalement sur l'applicabilité de chiffrement de certains éléments syntaxiques. La qualité d'images décodées est moyennement dégradée. Cependant, le point positif de cette étude est que l'approche est conforme au format de dernier standard. Hofbauer et al. [78] a proposé dans son approche un chiffrement transparent par la protection de bits de signes de ACs. Malheureusement, cette proposition est fausse car il n'existe plus des bits pour les signes ; les signes de chaque bloc sont codés ensemble en mode bypass. Les deux approches proposées par nous, permettent de remplacer les travaux de Shahid et al. [73][74]. Aussi, notre approches offrent le même niveau de sécurité visuelle qu'elles offrent les travaux de Shahid et al. sans changement de modèles de contextes.

Algorithme de chiffrement	C_1	C_2	C_3	C_4	C_5	C_6	C_7
Wallendael et al. [77]	Oui	Eléments syntaxiques (signs, MVDs, …)	Non	AES+XOR	Oui	Oui	Moyen
Hofbauer ***et al. [78]***	Oui	Bits de signes	Non	N/A	Oui	Non	Bas
Shahid et al. [73]	Oui	Codes et signes de QTCs	Oui	AES+CFB	Oui	Non	Haut
Shahid et al. [74]	Oui	Codes et signes de QTCs	Non	AES+CFB	Oui	Non	Haut
Approche #1 [76]	**Oui**	**Suffixes de code de Golomb-Rice + signes de QTCs**	**Non**	**AES-CBC**	**Oui**	**Oui**	**Haut**
Approche #2 [86]	**Oui**	**Les suffixes de codes de TR et de EG + signes de QTCs**	**Non**	**AES-CBC**	**Oui**	**Oui**	**Haut**

C_1: La préservation de la taille originale, C_2:les données chiffrées, C_3 :Context modeling, C_4 l'algorithme de chiffrement employé, C_5:Dépendance de compression. C_6: conformité de vidéo compressée. C_7: niveau de la dégradation visuelle.

Tableau VI.9 Etude comparative entre les approches proposées et les approches existantes pour le chiffrement de HEVC.

Notre deuxième approche [86] est une mise à niveau de [76]. En fait, elle est basée sur des bases théoriques solides contrairement à l'approche de [86] où nous avons référé au document standard de HEVC qui est été la seul documentation offerte au temps de travail. En conséquence, les résultats visuels obtenus dans [86] sont plus supérieures et meilleurs en termes de sécurité visuelle que ceux trouvés dans [76]. Aussi, [86] est plus rapide que [76] en temps d'exécution. L'avantage principal de [86] est qu'elle est applicable dans tous les niveaux de compression contrairement à [76] où elle est applicable seulement dans les hauts niveaux de compression.

VI.6 Conclusion

Dans ce chapitre, nous avons présenté une approche de chiffrement sélectif conforme au codage des données fréquentielle publié à la communauté scientifique.

Après avoir décrire le nouveau codage de QTCs et la manière dont ils sont binarise és, nous avons éclairci les nouveautés de ce codage par rapport aux solutions antérieurs, et la nécessité de la conception d'une nouvelle approche de chiffrement.

Nous avons montré à travers des tests expérimentaux le niveau de sécurité visuelle qu'elle offre notre approche. De même, nous avons vu que notre approche n'exige

plus assez de temps pour son exécution ce qui permet en outre, la déployer dans des applications multimédia diverses comme le streaming HD.

Le travail de ce chapitre constitue en effet, un premier pas vers la sécurité de vidéos HEVC. Où la norme HEVC est attendue à être le codec de l'ère Ultra HD dans la prochaine décennie.

Chapitre VII. Conclusion générale

Le but principal de cette thèse est de concevoir un système de crypto-compression pour la sécurité de l'information multimédia. Où nous avons choisi la vidéo car elle représente une information vive, dynamique et animée contrairement aux autres informations multimédia.

HEVC (High Efficiency Video Coding) est la dernière norme de codage vidéo, développée conjointement par les deux équipes de normalisation mondiaux IEC et ITU-T pour succéder la norme de codage H.264. Elle apporte une architecture hybride de codage permettant ainsi la compression de vidéos à haute définition. En effet, cette norme est attendue à être le futur codec de l'ère Ultra HD.

Comme la vidéo HEVC représente le résultat de compression d'une grande quantité de données visuelles. La préservation de la valeur estompée de cette norme par le maintien de la taille de flux binaire compressé s'avère inévitable lors de la conception d'un système de crypto-compression pour la protection de vidéos HEVC. Et ceci est atteint par l'inclusion de module de chiffrement durant l'étape de codage entropique de HEVC.

CABAC (Context-adaptive binary arithmetic coding) est le seul codeur entropique utilisé depuis HEVC. Le codage entropique des données fréquentielles surtout celui de QTCs a connu beaucoup de changement. En conséquence, les techniques antérieures de chiffrement de HEVC sont devenues non applicable et non conforme a la norme de HEVC.

Les travaux de contribution présentés dans cette thèse permettent de protéger les vidéos HEVC durant leur compression. Plus précisément, deux approches de chiffrement sélectif s'effectuant durant l'étape de codage entropique ont été présentées. D'abord, nous avons montré les limites observés dans les travaux antérieurs en termes de sécurité visuelle et de conformité selon le document standard de HEVC. Le chiffrement de QTCs durant le codage entropique est l'un des objectifs adressés dans certain approches comme celles de Shahid et al. Malheureusement, ces

approches sont conçues avant la normalisation de HEVC et elles nécessitent en conséquence une mise à niveau ou la recherche d'autres solutions alternatives permettant y remplacer. Pour relever ce défi, nous avons examiné le codage entropique de QTCs en se basant sur le document standard de HEVC qui nous à permis d'ouvrir nos yeux à la première approche. Où nous avons choisi à chiffrer les codes de Golomb-Rice qui ont été récemment introduits pour la binarisation de QTCs. Aussi, pour atteindre plus de dégradation visuelle, nous avons chiffrés aussi les signes de QTCs non nul. Le système de crypto-compression proposé utilise l'algorithme AES en mode de chainage CBC pour chiffrer les données sélectionnées. Dans une deuxième approche, nous avons reposé sur le dernier document bibliographique qui détaille le codage entropique de QTCs (la version finale de binarisation) contrairement à la première approche qui est basée seulement sur le format de flux binaire. La deuxième approche qu'on a proposée consiste à chiffrer de manière sûre et conforme les codes de QTCs selon le contexte de leur codage. Des qualités de sécurité comme la confidentialité et l'intégrité ont été bien sur assurées par nos deux approches. La confidentialité de nos approches est approuvée par l'utilisation d'algorithme robuste comme l'AES d'une part, et d'autre part par les résultats obtenus en termes de sécurités visuelle et de sécurité calculatoire. Aussi, la conformité et la taille originale de flux binaire crypté sont en effet des particularités émergentes qu'elles offrent nos approches.

Les travaux de contribution de cette thèse constituent véritablement un premier pas vers la sécurité de la norme HEVC. En effet, la qualité visuelle, le maintien de la taille de fichier compressé et la conformité de format offrent une grande motivation pour déployer les solutions proposées dans des applications multimédia diverses mobiles et réseaux, comme la diffusion en continu HD pour les chaines numériques, les réseaux sociaux, la vidéoconférence, et aussi la vidéosurveillance.

La sécurité de HEVC ne s'arrête plus dans les approches de chiffrement. Au contraire, elle s'étend aussi sur les techniques de stéganographie, de tatouage, et de la signature numérique. Le but de la stéganographie est de cacher des données de copyright à l'intérieur de flux binaire compressé de HEVC. De même, le tatouage permet quant à lui d'insérer une marque dans le flux binaire afin de protéger les droits de propriétaire.

L'utilisation d'autres cryptosystèmes robustes est un autre axe de recherche qu'on peut suggérer pour les futures recherches. Les approches basées sur le chaos sont en effet des solutions prometteuses permettant ainsi d'offrir un niveau de sécurité satisfaisant, car elles sont très sensibles aux paramètres initiaux qui constituent souvent les clés de systèmes cryptographiques.

Aussi, la sécurité d'extensions de la norme HEVC multivue et scalable constitue en effet un autre art de recherche, car elles seront déployées dans des applications diverses multimédia comme les applications de vision 3D. De même, la sécurité d'autres formats de l'information multimedia comme l'image et le son sera quant à elle un vif sujet suggéré pour la recherche surtout avec l'expansion de formats de haute définition.

Chapitre VIII. Références bibliographiques

[1] “Larousse.fr : encyclopédie et dictionnaires gratuits en ligne.” [Online]. Available: http://www.larousse.fr/. [Accessed: 29-May-2015].

[2] “Gravures rupestres du Tassili,” *Wikipédia*. [Online]. Available: https://fr.wikipedia.org/wiki/Gravures_rupestres_du_Tassili/. [Accessed: 29-May-2015].

[3] C. E. Shannon, “A Mathematical Theory of Communication,” *Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, Jul. 1948.

[4] D. A. Huffman, “A Method for the Construction of Minimum-Redundancy Codes,” *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, Sep. 1952.

[5] J. Rissanen and G. G. Langdon, “Arithmetic Coding,” *IBM J. Res. Dev.*, vol. 23, no. 2, pp. 149–162, Mar. 1979.

[6] D. Salomon, G. Motta, and D. Bryant, *Handbook of Data Compression*, 5th ed. 2010. London ; New York: Springer London Ltd, 2009.

[7] V. S. Miller and M. N. Wegman, “Variations on a theme by Ziv and Lempel,” in *Combinatorial Algorithms on Words*, A. Apostolico and Z. Galil, Eds. Springer Berlin Heidelberg, 1985, pp. 131–140.

[8]M. Hassoun, *Fundamentals of Artificial Neural Networks*. Cambridge, Mass.: A Bradford Book, 2003.

[9] N. S. Altman, “An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression,” *The American Statistician*, vol. 46, no. 3, pp. 175–185, Aug. 1992.

[10]Y. Linde, A. Buzo, and R. M. Gray, “An Algorithm for Vector Quantizer Design,” *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, Jan. 1980.

[11] “A simple introduction to the KLT (Karhunen—Loève Transform),” in *Deep Space Flight and Communications*, Springer Berlin Heidelberg, 2009, pp. 151–179.

[12] M. Bellanger, *Traitement numérique du signal - 9e éd.*, 9e édition. Paris: Dunod, 2012.

[13] N. Ahmed and K. R. Rao, “Walsh-Hadamard Transform,” in *Orthogonal Transforms for Digital Signal Processing*, Springer Berlin Heidelberg, 1975, pp. 99–152.

[14] A. N. Akansu and P. R. Haddad, *Multiresolution Signal Decomposition, Second Edition: Transforms, Subbands, and Wavelets*, 2 edition. San Diego: Academic Press, 2000.

[15] N. Ahmed, T. Natarajan, and K. R. Rao, “Discrete Cosine Transform,” *IEEE Transactions on Computers*, vol. C–23, no. 1, pp. 90–93, Jan. 1974.

[16] B. Furht, “A Survey of Multimedia Compression Techniques and Standards. Part I: JPEG Standard,” *Real-Time Imaging*, vol. 1, no. 1, pp. 49–67, Apr. 1995.

[17] Standard MPEG-1 : ISO/IEC 11172-2, Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s.

[18] Standard MPEG-2 : ISO/IEC 13818-2, Information Technology – Generic coding of moving pictures and associated audio information.

[19] Standard MPEG-4 : ISO/IEC 14496-2, Information Technology – Coding of Audio-Visual Objects.

[20] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[21] B. D. Tseng and W. C. Miller, "On Computing the Discrete Cosine Transform," *IEEE Transactions on Computers*, vol. C–27, no. 10, pp. 966–968, Oct. 1978.

[22] http://www.w3.org/Graphics/GIF/spec-gif89a.txt [Accessed: 29-May-2015].

[23]http://www.libpng.org/pub/png/[Accessed: 29-May-2015].

[24] http://www.jpeg.org/jpegls/index.html[Accessed: 29-May-2015].

[25] http://www.jpeg.org/jpeg2000/index.html [Accessed: 29-May-2015].

[26] http://fr.wikipedia.org/wiki/Motion_JPEG [Accessed: 29-May-2015].

[27] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, décembre 2012.

[28] *Weber's Law of Just Noticeable Difference*, University of South Dakota: http://apps.usd.edu/coglab/WebersLaw.html

[29] N.-M. Nguyen, X.-T. Tran, P. Vivet, and S. Lesecq, "An efficient Context Adaptive Variable Length coding architecture for H.264/AVC video encoders," in *2012 International Conference on Advanced Technologies for Communications (ATC)*, 2012, pp. 158–164.

[30] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 620–636, Jul. 2003.

[31] D. X. Tian, D. T. M. Le, and P. D. Y. Lian, "Review of CAVLC, Arithmetic Coding, and CABAC," in *Entropy Coders of the H.264/AVC Standard*, Springer Berlin Heidelberg, 2011, pp. 29–39.

[32] H.261 : Video codec for audiovisual services at p x 64 kbit/s. Recommandation H.261 à l'UIT-T. Mars 1993.

[33] H.263 : Video coding for low bit rate communication. Première Recommandation H.263 à l'UIT-T. Mars 1996.

[34] H.263+ : Video coding for low bit rate communication. Deuxième recommandation H.263 à l'UIT-T. Février 1998.

[35] H.263++ : H.263 Annex U, V, W and X. Compléments de la recommandation H.263 à l'UIT-T. Janvier 2000.

[36] ITU-T VCEG and ISO/IEC MPEG, "Joint Call for Proposals on Video Compression Technology", document VCEG-AM91 of VCEG and N11113 of MPEG, (2010).

[37] High efficiency video coding, ITU-TRec.H.265 and ISO/IEC 23008-2 (MPEG-H, Part 2), Apr.(2013), version 1.

[38] G. J. Sullivan, and J.-R. Ohm, “Meeting report of the first meeting of the Joint Collaborative Team on Video Coding (JCT-VC), Dresden, DE, 15-23 April, 2010”, document JCTVC-A200 of JCT-VC, (2010).

[39] JCT-VC, “WD1: Working Draft 1 of High-Efficiency Video Coding”, JCTVC-C403, JCT-VC Meeting, Guangzhou, October 2010.

[40] Y. H. Tan, C. Yeo, H. L. Tan, and Z. Li, “On residual quad-tree coding in HEVC,” in *2011 IEEE 13th International Workshop on Multimedia Signal Processing (MMSP)*, 2011, pp. 1–4.

[41] V. Sze and M. Budagavi, “High Throughput CABAC Entropy Coding in HEVC,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1778–1791, décembre 2012.

[42] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C*, 2nd edition. New York: Wiley, 1996.

[43] M. J. Dworkin, “SP 800-38A Addendum. Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode,” National Institute of Standards & Technology, Gaithersburg, MD, United States, 2010.

[44] M. Khan and T. Shah, “A Literature Review on Image Encryption Techniques,” *3D Res.*, vol. 5, no. 4, pp. 29:1–29:25, Dec. 2014.

[45] X. Jia, “Image Encryption Using the Ikeda Map,” in *2010 International Conference on Intelligent Computing and Cognitive Informatics (ICICCI)*, 2010, pp. 455–458.

[46] A. P. Tafti and S. Janosepah, “Digital Images Encryption in Frequency Domain Based on DCT and One Dimensional Cellular Automata,” in *Informatics Engineering and Information Science*, A. A. Manaf, A. Zeki, M. Zamani, S. Chuprat, and E. El-Qawasmeh, Eds. Springer Berlin Heidelberg, 2011, pp. 421–427.

[47] M. Abomhara, O. Zakaria, O. O. Khalifa, A. . Zaidan, and B. . Zaidan, “Enhancing Selective Encryption for H.264/AVC Using Advanced Encryption Standard,” *International Journal of Computer and Electrical Engineering*, pp. 223–229, 2010.

[48] D. Coppersmith, “The Data Encryption Standard (DES) and Its Strength Against Attacks,” *IBM J. Res. Dev.*, vol. 38, no. 3, pp. 243–250, May 1994.

[49] J. Daemen and V. Rijmen, *The Design of Rijndael: AES - The Advanced Encryption Standard*, Softcover reprint of the original 1st ed. 2002 edition. Berlin, Heidelberg: Springer, 2013.

[50] Barker, C.William; Barker, Elaine (January 2012). "NIST Special Publication 800-67 Revision 1: Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher.

[51] R. L. Rivest, A. Shamir, and L. Adleman, “A Method for Obtaining Digital Signatures and Public-key Cryptosystems,” *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[52] T. Elgamal, “A public key cryptosystem and a signature scheme based on discrete logarithms,” *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, Jul. 1985.

[53] Ralph Merkle and Martin Hellman, Hiding Information and Signatures in Trapdoor Knapsacks, *IEEE Trans. Information Theory*, 24(5), September 1978, pp525–530.

[54] M. Pazarci and V. Dipcin, “A MPEG2-transparent scrambling technique,” *IEEE Transactions on Consumer Electronics*, vol. 48, no. 2, pp. 345–355, May 2002.

[55] D. Socek, S. Magliveras, D. Ćulibrk, O. Marques, H. Kalva, and B. Furht, "Digital Video Encryption Algorithms Based on Correlation-Preserving Permutations," *EURASIP Journal on Information Security*, vol. 2007, no. 1, p. 052965, Jul. 2007.

[56] S. Li, G. Chen, A. Cheung, B. Bhargava, and K.-T. Lo, "On the Design of Perceptual MPEG-Video Encryption Algorithms," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 2, pp. 214–223, Feb. 2007.

[57] J Meyer, F. Gadegast, Security mechanisms for multimedia data with the example MPEG-1 video, project description of SECMPEG. Technical University of Berlin; 1995.

[58] U. Potdar, K. T. Talele, and S. T. Gandhe, "Comparison of MPEG Video Encryption Algorithms," in *Proceedings of the International Conference on Advances in Computing, Communication and Control*, New York, NY, USA, 2009, pp. 289–294.

[59] W. Zeng and S. Lei, "Efficient frequency domain selective scrambling of digital video," *IEEE Transactions on Multimedia*, vol. 5, no. 1, pp. 118–129, Mar. 2003.

[60] L. Tang, "Methods for Encrypting and Decrypting MPEG Video Data Efficiently," in *Proceedings of the Fourth ACM International Conference on Multimedia*, New York, NY, USA, 1996, pp. 219–229.

[61] C. Shi, S. Wang, and B. Bhargava, "MPEG Video Encryption in Real-time Using Secret Key Cryptography," in *in Proc. Int. Conf. Parallel and Distributed Processing Techniques and Applications*, 1999.

[62] C. Wu and C. –. J. Kuo, "Design of integrated multimedia compression and encryption systems," *IEEE Trans. Multimedia*, vol. 7, pp. 828–839, 2005.

[63] S. Li, G. Chen, A. Cheung, B. Bhargava, S. Li, G. Chen, A. Cheung, B. Bhargava, and K. Lo, "On the design of perceptual MPEG-video encryption algorithms," *Humidity", Aerospace Recommended Practice No. 866A, SAE Inc., 400 Commonwealth Drive, Warrendale, PA 15096*, pp. 214–223, 2005.

[64] C. Wu and C. –. J. Kuo, "Design of integrated multimedia compression and encryption systems," *IEEE Trans. Multimedia*, vol. 7, pp. 828–839, 2005.

[65] J. Wang, Y. Fan, T. Ikenaga, and S. Goto, "A partial scramble scheme for H.264 video," in *7th International Conference on ASIC, 2007. ASICON '07*, 2007, pp. 802–805.

[66] D. Wang, Y. Zhou, D. Zhao, and J. Mao, "A Partial Video Encryption Scheme for Mobile Handheld Devices with Low Power Consideration," in *International Conference on Multimedia Information Networking and Security, 2009. MINES '09*, 2009, vol. 2, pp. 99–104.

[67] T. D. B. Weerasinghe, "An effective RC4 stream cipher," in *2013 8th IEEE International Conference on Industrial and Information Systems (ICIIS)*, 2013, pp. 69–74.

[68] H. Sohn, E. T. AnzaKu, W. De Neve, Y. M. Ro, and K. N. Plataniotis, "Privacy Protection in Video Surveillance Systems Using Scalable Video Coding," in *Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, 2009. AVSS '09*, 2009, pp. 424–429.

[69] H.-J. Lee and J. Nam, “Low Complexity Controllable Scrambler/Descrambler for H.264/AVC in Compressed Domain,” in *Proceedings of the 14th Annual ACM International Conference on Multimedia*, New York, NY, USA, 2006, pp. 93–96.

[70] Y. Kim, S. H. Jin, T. M. Bae, and Y. M. Ro, “A selective video encryption for the region of interest in scalable video coding,” in *TENCON 2007 - 2007 IEEE Region 10 Conference*, 2007, pp. 1–4.

[71] Z. Shahid, M. Chaumont, and W. Puech, “Fast Protection of H.264/AVC by Selective Encryption of CAVLC and CABAC for I and P Frames,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 5, pp. 565–576, mai 2011.

[72] M. N. Asghar, M. Ghanbari, M. Fleury, and M. J. Reed, “Sufficient encryption based on entropy coding syntax elements of H.264/SVC,” *Multimed Tools Appl*, pp. 1–27, Jul. 2014.

[73] Z. Shahid and W. Puech, “Visual Protection of HEVC Video by Selective Encryption of CABAC Binstrings,” *IEEE Transactions on Multimedia*, vol. 16, no. 1, pp. 24–36, Jan. 2014.

[74] Z. Shahid and W. Puech, “Investigating the structure preserving encryption of high efficiency video coding (HEVC),” 2013, vol. 8656, p. 86560N–86560N–10.

[75] HEVC, “High Efficiency Video Coding (HEVC) Text Specification Draft 6,” Tech. Rep., ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), San Jose, CA, USA, 2012, Doc. JCTVC-H1003.

[76] Mokhtar Ouamri and Kamel Mohamed Faraoun, ‘’ Robust and fast selective encryption for HEVC videos’’, JOURNAL OF COMMUNICATIONS SOFTWARE AND SYSTEMS, VOL. 10, NO. 4, DECEMBER 2014, pp. 221–229.

[77] G. Van Wallendael, A. Boho, J. De Cock, A. Munteanu, and R. Van de Walle, “Encryption for high efficiency video coding with video adaptation capabilities,” *IEEE Transactions on Consumer Electronics*, vol. 59, no. 3, pp. 634–642, Aug. 2013.

[78] H. Hofbauer, A. Uhl, and A. Unterweger, “Transparent encryption for HEVC using bit-stream-based selective coefficient sign encryption,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1986–1990.

[79] J. Daemen and V. Rijmen, *The Design of Rijndael: AES - The Advanced Encryption Standard*, Softcover reprint of the original 1st ed. 2002 edition. Berlin, Heidelberg: Springer, 2013.

[80] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[81] L. Dubois, W. Puech, and J. Blanc-Talon, “Reduced selective encryption of intra and inter frames of H.264/AVC using psychovisual metrics,” in *2012 19th IEEE International Conference on Image Processing (ICIP)*, 2012, pp. 2641–2644.

[82] J. Sole, R. Joshi, N. Nguyen, T. Ji, M. Karczewicz, G. Clare, F. Henry, and A. Duenas, “Transform Coefficient Coding in HEVC,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1765–1777, décembre 2012.

[83] T. Nguyen, P. Helle, M. Winken, B. Bross, D. Marpe, H. Schwarz, and T. Wiegand, "Transform Coding Techniques in HEVC," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 978–989, décembre 2013.

[84] T. Nguyen, P. Helle, M. Winken, B. Bross, D. Marpe, H. Schwarz, and T. Wiegand, "Corrections to;Transform Coding Techniques in HEVC";," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 6, pp. 1194–1195, décembre 2014.

[85] F.Bossen, "Common HM test conditions and software reference configurations," document JCTVC-L1100 of JCT-VC, Geneva, CH, Jan. 2013.

[86] Mokhtar Ouamri and Kamel Mohamed Faraoun, '' New Compliant Scheme for Selective Encryption of HEVC/H.265 Videos'', Journal of Information Processing Systems (ISSN: 1976-913X(Print), ISSN: 2092-805X(Online)) .

Robust and fast selective encryption for HEVC videos

Mokhtar Ouamri, and Kamel Mohamed Faraoun

Abstract — **Emerging High efficiency video coding (HEVC) is expected to be widely adopted in network applications for high definition devices and mobile terminals. Thus, construction of HEVC's encryption schemes that maintain format compliance and bit rate of encrypted bitstream becomes an active security's researches area. This paper presents a novel selective encryption technique for HEVC videos, based on enciphering the bins of selected Golomb–Rice code's suffixes with the Advanced Encryption Standard (AES) in a CBC operating mode. The scheme preserves format compliance and size of the encrypted HEVC bitstream, and provides high visual degradation with optimized encryption space defined by selected Golomb–Rice suffixes. Experimental results show reliability and robustness of the proposed technique.**

Index Terms — **High efficiency video coding, Golomb–Rice code, Context-adaptive binary arithmetic coding, advanced encryption system.**

I. INTRODUCTION

High efficiency video coding (HEVC) is the latest video coding standard [1] developed by Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) as a successor of H.264/AVC [2]. One of its primary objectives is to provide almost double compression efficiency at the cost of major computational complexity increase with respect to its predecessor H.264/AVC. It also support wide range of high definition video resolutions (from Full HD 1920x1080 to 4K Ultra HD and 8K Ultra HD) and several corresponding frame rates (30 FPS to 120 FPS).

The HEVC coding efficiency is optimized by improving the core of basic hybrid coding architecture of its predecessor H264/AVC by introducing several features and tools in all mains stages of compression including prediction, transformation, quantization, and entropy coding. Due to its decent coding performance, the emerging HEVC standard is expected to be widely adopted in network applications for HD devices and mobile terminals [3] such as ultra high-definition television UHDTV, streaming, and low delay communication. Security of such video applications is based on the protection of communicated HEVC videos using efficient encryption techniques, according to one of two possible encryption modes: a full encryption mode applying a global ciphering of the HEVC bitstream in detriment of the format compliance of the standard, and a selective encryption mode that only encipher some selected parts of the video data (transform coefficients, signs of motion vectors, syntax elements of entropy coder,…) without destroying the format compliance of the HEVC standard. In addition to HEVC format compliance, the selective encryption mode ensures the same bit rate ratio of encrypted bitstream as the original bitstream.

In order to apply selective encryption mode, a retrieval of the meaningful data to be encrypted must firstly be performed in order to get maximum visual degradation of the encrypted video sequences. Since the transform coefficients are the most widely employed as protected parts for selective encryption mode, we propose in this work to use the Golomb–Rice codes newly defined by the HEVC standard as selected parts to be protected.

The remaining of this paper is organized as follows: a brief description of HEVC structure with corresponding coding tools, coefficient level coding, and related encryption works are presented in Section II. Section III explain the procedure of level plaintext preparing and encryption/decryption. Section IV presents the different performed experiments with obtained result. Finally, conclusions are drawn in section V.

II. BACKGROUND AND RELATED WORKS

A. Description of HEVC structure and coding tools

The emerging high efficiency video coding (HEVC) is a new video coding standard, which introduces several tools and new concepts for a better compression of multiple existing picture resolutions, chiefly high definition (HD) and ultra high definition UHD spatial resolutions. To ensure a highest level of compression efficiency, the input video frame is first split into multiple coding-tree units (CTUs) with a maximum size of 64x64 pixels. A CTU is the basic unit of coding process and can be part of a slice (I_SLICE, P_SLICE, and B_SLICE) or a tile. Every CTU is a root of a quadtree structure that can be later divided into leaf level coding units (CUs) sharing the same mode of prediction. Each CU is partitioned further into prediction units (PUs), and can either uses intra-frame prediction (a whole of 35 modes are available) or inter frame prediction (uni-prediction or bi-prediction). The residual coding is done by partitioning each CU in a quadtree, and then defining the transform unit (TU) as a leaf of CU quadtree structure. The TU is the basic unit used for transform and quantization stage, and it has a dyadic block size varying from 4x4 to 32x32 samples. In addition to CTU, CU, PU, and TU,

Manuscript received May 25, 2014; revised December 15, 2014.

M. Ouamri is with the Department of Computer science, Djilalli Liabbes University, Sidi Bel Abbès, Algeria. (e-mail: amokhtar124@yahoo.fr).

K,M. Faraoun is with the Department of Computer science, Djilalli Liabbes University, Sidi Bel Abbès, Algeria. (e-mail: kamel_mh@yahoo.fr).

1845-6421/12/8359 © 2014 CCIS

various new features are introduced in different stages of encoding/decoding operation of HEVC standard, and are explicitly highlighted in [1].

The CABAC (Context adaptive binary arithmetic coding) is the only one standard supported for entropy coding [3] in the HEVC, and it is an improved and simplified straightforward extension of the CABAC used in H.264/AVC standard [4]. The three main operations involved by the CABAC engine are depicted in Fig. 1: a binarization to decompose the non-binary syntax elements into a sequence of bins, a context modeling, and a binary arithmetic coding (BAC).Two operation modes are invoking from CABAC engine: a regular mode where the context model is required for coding bins, and a bypass mode where bins are coded with equi-probability. Many techniques were added to improve the throughput [3], including reducing context coded bins, grouping bypass bins together, grouping bins that use the same contexts together, reducing context selection dependencies, and reducing the total number of signaled bins. Furthermore, the HEVC define a new set of CABAC coded syntax elements for describing the properties of CU, PU, TU, and Loop filter. The draft of HEVC [6] presents several different binarization processes including unary coding, truncated unary coding, k^{th} order Exp-Golomb (EG_k) coding, Golomb–Rice coding and fixed length coding.

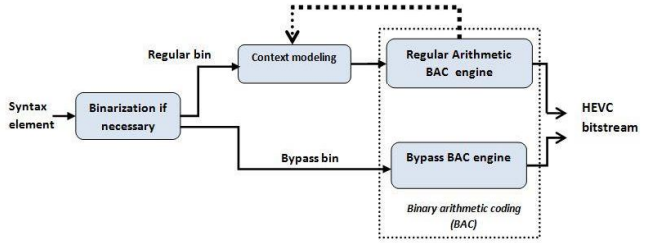


Fig. 1. Bloc diagram of CABAC engine with three key operations: binarization, context modeling and binary arithmetic coding (BAC).

B. Overview of coefficient level coding

A transform unit of size N x M consists of at least N x M quantized transform coefficients. The transform blocks TBs for TUs larger than 4x4 are decomposed into 4x4 sub-blocks unit, when each sub-block contains 16 consecutive quantized coefficients, encoded in inverse diagonal scan order. To encode coefficients level of each sub-block, five syntax elements are used to represent the coefficient's level information within the sub-block if it contains one or more non zero quantized transform coefficients: significant_coeff_flag, coeff_abs_level_greater1_flag, coeff_abs_level_greater2_flag, coeff_sign_flag, and coeff_abs_level_remaining. Table I describes the semantic of each cited syntax element.

Adaptive context models through regular mode is employed for encoding significant_coeff_flag, coeff_abs_level_greater1_flag and coeff_abs_level_greater2_flag, when remaining syntax elements coeff_sign_flag and coeff_abs_level_remaining are encoded by low complexity bypass mode.

The HEVC utilizes only Golomb–Rice code and k^{th} order Exp-Golomb (EG_k) for coeff_abs_level_remaining binarization [7] as depicted in Fig. 2. A Golomb–Rice code is an optimal code for representing a symbol value n and is defined by the quotient q=[n/m] and the remainder p=n-q.m, whereas m is a rice parameter. Quotient represents the prefix part binarized with a unary code, and remainder represents the suffix part composed by a fixed length bins. The Exp-Golomb code of symbol value n is obtained by concatenation of prefix and suffix codeword. The prefix is the unary code of $l(n)=\log_2((n/2^k)+1)$, whereas the suffix is calculated by $n+2^k(1-2^{l(n)})$.

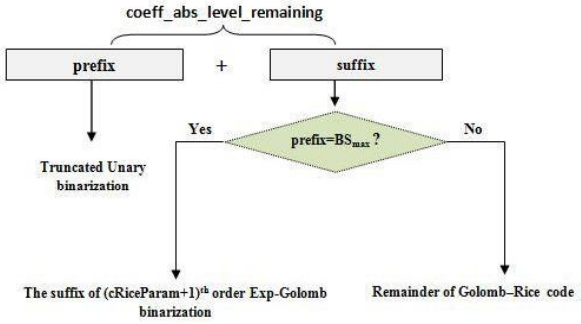


Fig. 2. Binarization of coeff_abs_level_remaining : prefix is binarized with truncated unary code, and suffix is binarized either by Golomb–Rice code or by Exp-Golomb code

TABLE I
DESCRIPTION OF CABAC SYNTAX ELEMENTS EMPLOYED FOR LEVEL CODING

Syntax element	Description
significant_coeff_flag	indicates the significance of each coefficient
coeff_abs_level_greater1_flag	indicates whether the coefficient amplitude is larger than one for each non zero coefficient
coeff_abs_level_greater2_flag	indicates whether the coefficient amplitude is larger than two for each coefficient with amplitude larger than one
coeff_sign_flag	indicates sign information of the nonzero coefficients
coeff_abs_level_remaining	indicates remaining absolute level value

According to [6] and [7], binarization of coeff_abs_level_remaining consists of a prefix part and a suffix part, and it depends on two parameters: cRiceParam and cTRMax. The cRiceParam parameter is ranging from 0 to 4 and it changes depending on the previously coded coeff_abs_level_remaining. Firstly, the truncated unary binarization is invoked to derive the prefix binstrings by binarizing the part Min(4, coeff_abs_level_remaining >> cRiceParam). If the prefix bin string is equal to a predefined bit string noted BS_{max}, then the suffix bin string is derived using the Exp-Golomb binarization for the suffix part (coeff_abs_level_remaining - cTRMax) with an order set equal to cRiceParam + 1, otherwise, the suffix string is the remainder of Golomb–Rice coding process specified by the

binary representation of the value computed by: (coeff_abs_level_remaining- ((coeff_abs_level_remaining>> cRiceParam) << cRiceParam) with a fixed length equal to cRiceParam.

The HEVC employs Golomb–Rice codes for short symbol's values, and Exp-Golomb codes for representing long symbol values. The Coefficients coeff_sign_flag are regrouped and encoded together for each sub-block, and are signaled before the coeff_abs_level_remaining of non zeros coefficients.

C. Related works

Selective encryption is a new trend for format-compliant content protection, privacy and security. It consists to choose a subset of data as protected part to be encrypted, and to let the remaining data unencrypted as a public part. The last decade is characterized by a number of important works on video selective encryption.

Various schemes have been suggested in compressed video especially for the H.264/AVC standard which uses CABAC and CAVLC for entropy coding. According to the protected part selected to be encrypted, the works proposed in [8] and [9] encrypt the DCT coefficient syntax elements (Non-zeros level and signs), while those in [10] and [11] scramble the intra mode prediction and the motion vector difference.

In [12] , the author proposes one of the first works allowing secure HEVC encoding by selective encryption. His works adopts a realization of HEVC level encoding with a binarization of coeff_abs_level_remaining using Golomb–Rice code for short symbol, and with zeroth Exp-Golomb (EG_0) for long symbol. The work focuses firstly on searching a set of dyadic codes of coeff_abs_level_remaining that can be encrypted without altering the format compliance of the HEVC bitstream. Then, the author prepares a plaintext formed by dyadic codes of coeff_abs_level_remaining and levels sign, and encrypted the resulting using AES-CFB mode. Unfortunately , since coeff_abs_level_remaining is binarized as described in [6], the scheme proposed in employment of Zafar's technique [12] cannot be used with the latest realization of the HEVC standard.

Several recent works deal with HEVC encryption using different approaches. In [13], the author encrypt three selected bitstream elements, namely intra prediction mode difference, motion vector difference sign, and residual sign. In [14], the residue data, intra-prediction modes, inter-prediction modes and motion vectors are key elements that are selected to encrypt to keep the security. Similar works can also be found in the works [15-17].

In this work, we present an original technique for selective encryption allowing the protection of HEVC video, and permitting to preserve the format compliance of HEVC bitstream.

III. The Proposed Approach

Among the five syntax elements used for representing each 4x4 sub-block coefficients of the transform unit, only coeff_sign_flag and coeff_abs_level_remaining suffixes can be used to form the encryption space. Exp-Golomb code and Golomb–Rice code are utilized for binarizing coeff_abs_level_remaining, when the first one is employed to encode less frequent symbols with an order equal to cRiceParam+1, and the Golomb–Rice code is widely used to encode symbols having high probability of occurrence and is the most demanded by HEVC encoder for increasing compression ratio. The Golomb-Rice's suffix is an optimal code formed by a binary representation with length equal to cRiceParam bins (for example if cRiceParam is equal to 3, the binary representation of suffix is formed by three bins) .When cRiceParam value is greater or equal to 1, any modifications that affect the suffix binary representation through the same fixed length of bins will not alter the format compliance of Golomb–Rice code because here suffix means merely a remainder of division having the same fixed length of bins. Consequently, we defines the encryption space ES by the set of different binary representations of the Golomb–Rice suffixes of coeff_abs_level_remaining having length greater or equal to 1.

In order to preserve the format compliance of HEVC bitstream, it is necessary to avoid modifying DCs coefficients of TUs, since they represent the average of the TUs energy, and it is preferred to obviate the first and the last coefficient of each reverse scan pattern in each sub-block especially when TUs range in size from 8x8 to 32x32.

It is generally preferred to secure HEVC video only in low delay mode employing a particular order for decoding process, since in other modes, the encryption in random access scheme can alter the format compliance as the HEVC decoder choose any part of the frame at random.

The principle aim of the proposed technique is to keep the format compliance and the bit rate of the encrypted bitstream without any alterations. The decoded frames of the encrypted bitstream must have high visual degradation compared to the original plain frames. The protected parts chosen in proposed selective encryption technique are defined by the set of elements belonging to the encryption space ES.

A. Preparing of plaintext of coefficients levels

For each intra predicted slice I_SLICE, if a transform unit TU is divided into N sub-block named sub-$block_1$,sub-$block_2$,… , sub-$block_N$ consecutively, then only sub-$block_1$ can contain DC coefficient. In addition, if the reverse scanning order in each sub-block starts with a significant coefficient noted $Coef_1$ and proceeds to another significant coefficient noted $Coef_2$ ($Coef_2$ can be DC of TU in sub-$block_1$), then we can encrypt all coefficients of each sub-block except the two coefficients $Coef_1$ and $Coef_2$. Selected coefficients will be denoted further by ACEs.

Fig. 3 shows an example of a TU_{8x8} decomposed into 4 sub-block. In this example, we suppose that all coefficients are non zeros, so we can encrypt all coefficients except those colored in white. The binary plaintext noted plaintext1 is constructed by concatenating all bins of coeff_abs_level_remaining suffixes of the ACEs belonging to ES. This is done by appending into a plaintext P_1 all cRiceParam bins of the binary

representation of each Golomb–Rice suffix found when cRiceParam value is greater or equal to 1.

The number of Golomb–Rice suffixes of coeff_abs_level_remaining syntax elements qualified to be encrypted varies from one 4x4 sub-block to another, and there is at least 14 elements per sub-block. If a sub-block contains L Golomb–Rice suffixes, and if we define L_{MAX} as a number non zero less than L, then any change affects the first L_{MAX} Golomb–Rice suffixes in inverse scan order will modify the whole sub-block entirely. For such reason, we optimized the plaintext P_1 by choosing the bins belonging to the first L_{MAX} Golomb–Rice suffixes found for each 4x4 sub-block, when the range of L_{MAX} ranges from 1 to 14 and defined as a secret parameter.

B. Encryption and decryption procedure

For all intra predicted slice (I_SLICE), we encrypted its corresponding constructed plaintext P_1 in a CBC mode [18]. First we divided the plaintext into a sequence of n+1 consecutives blocks (X_1, X_2,...., X_n, Y) where $n \geq 0$. The length of each block X_i is fixed to 128bit, whereas the length of remainder plaintext Y can be less than 128. A random initial vector $IV \in \{0,1\}^{128}$ is then chosen at random, and the first ciphered block C_1 is generated by applying the AES cipher to $X_1 \oplus IV$ (denotes or-exclusive bit to bit operation) using a secret key K_1 with having a length of 128bits. For i=2...n, each remaining ciphered block C_i is obtained by encrypting $C_{i-1} \oplus X_i$ using AES and the key K_1. The latest block cipher C_{n+1} is specially computed by $C_{n+1}=Y \oplus AES(K_2,IV)$, when K_2 is a secret key on 128 bits ,that it different from K_1 to keep high privacy of the CBC-AES mode. Finally, the ciphertext is obtained by concatenating the blocks C_1, C_2,...., C_n, C_{n+1}.

After encrypting the blocks, selected suffix bin of the plaintext are replaced by correspondent bin in the ciphertext before the BAC compression. The decryption can be performed after BAC decompression by the deciphering suffixes bins of the decoded ciphertext of the. Each ciphertext block is then replaced by its corresponding decrypted plaintext bins.

Since the proposed scheme encipher only Golomb-Rice suffixes, the format of the encoded video sequence is preserved. In addition, since encryption using AES-CBC mode preserves the size of the plaintext, the scheme also preserves the size of encrypted video frames. In the next section, several experiments and performances evaluations are performed on the proposed scheme, with comparisons to some existing video encryption schemes.

IV. EXPERIMENTS AND OBTAINED RESULTS

The proposed scheme is implemented and tested using the HEVC reference software HM v10.0 [19]. In all experimental tests, encryption/decryption are performed simultaneously with encoding/decoding by embedding corresponding modules into the reference software. Simulation results described in this section were processed on benchmark video sequences of different sizes including WQVGA(416×240), WVGA(832×480), SD(1280×720), HD(1920×1080), UltraHD(2560×1600) and 4K UHD(3840×2160) illustrated in Table II with corresponding frame-rates. Table III shows the common encoding parameters used in all tested modes of the reference software (low delay and random access).

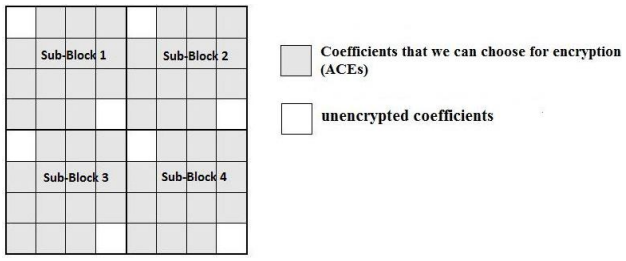


Fig. 3. Transform unit T_{8x8} with 4 sub-block: gray cases mean selected coefficients for encryption.

TABLE.II.
BENCHMARK VIDEO SEQUENCES USED TO SIMULATE THE PROPOSED SCHEME

Video sequence	Resolution	Frame-rate
BasketballPass	416×240	50
BasketballDrill	832×480	50
Johnny	1280×720	60
BasketballDrive	1920×1080	50
Traffic	2560×1600	30
YachtRide	3840×2160	120

TABLE.III.
THE SET OF ENCODER PARAMETERS USING DURING EXPERIMENTS WITH CORRESPONDING VALUES

Parameter	Description	Used value
MaxCUWidth	Maximum coding width in pixel	64
MaxCUHeight	Maximum coding height in pixel	64
MaxPartitionDepth	Maximum coding unit depth	4
IntraPeriod	Intra frame period	8
GOPSize	Size of the GOP's structure	8
InputBitDepth	8 bit per pixel	8

A. Parameters evaluation

The first experiment was performed in low delay mode, where each GOP is composed from an intra frame (I frame) followed by 7 predicted frame (P frame). Fig. 4 shows the first frame of the original benchmark sequences without encryption, and Fig. 5 shows the decoded frames at a quantization parameter QP=18 after encryption with secret parameter L_{MAX}=14. It is apparent that the commercial values of decoded frames are fully destroyed with high visual degradation, and perceptual content is completely disguised. Fig. 6 shows the 9th, 11th, and 14th original frames of BasketballDrill sequence encoded and encrypted at QP=24 with secret parameter L_{MAX}=14, with the corresponding decoding results. Only the 9th frame is encrypted because it is and intra-frame whereas 11th and 14th frames are P frames. It is clear from illustrated results that the visual protection of intra

intra frame propagates to its following P frames.

Fig. 4. The first original frame of: (a) BasketballPass,(b) BasketballDrill,(c) Johnny,(d) BasketballDrive,(e) Traffic and (f) YachtRide

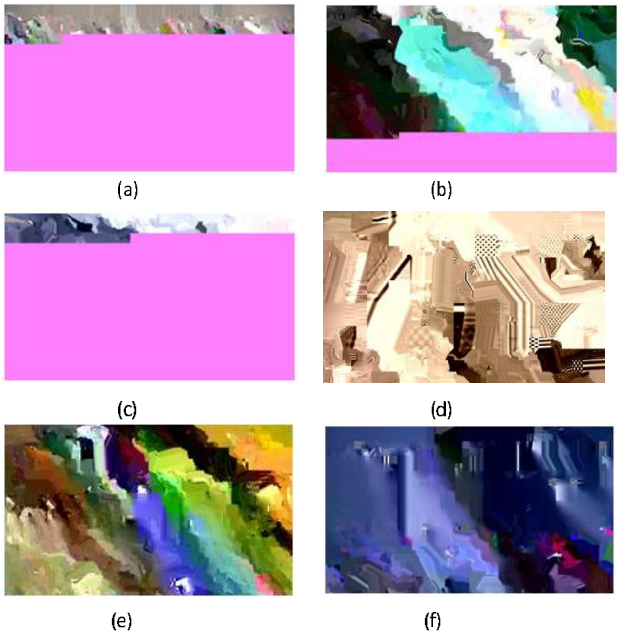


Fig.5. First decoded frame without decryption at QP=18 and L_{MAX}=14 : (a) BasketballPass, (b) BasketballDrill, (c) Johnny, (d)BasketballDrive, (e)Traffic and (f)YachtRide.

In a second experiment, we encrypted and encoded the first frame of BasketballDrill sequence at QP=24 while testing different values of L_{MAX} . Fig. 7 shows original frame used for encryption, with corresponding decoded frames (without decryption) when the secret encryption parameter L_{MAX} take the values 4, 8, and 12 respectively. Decoding results changes according to L_{MAX} , and we note that sufficient visual degradation is obtained for L_{MAX}=4, and all illustrated results prove that decoded frames don't share any outline objects with original ones.

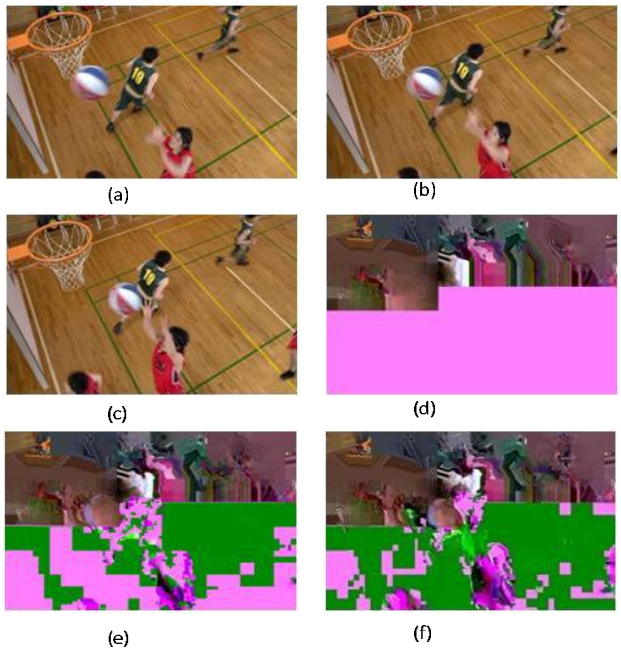


Fig. 6. Original and decoded frames without decryption at QP=24 and L_{MAX}=14 for the BasketballDrill sequence: (a),(b)and (c) are the 9th, 11th,and 14th original frames respectively, and (d),(e),and (f) are the 9th, 11th,and 14th encrypted frames decoded as I,P,P respectively.

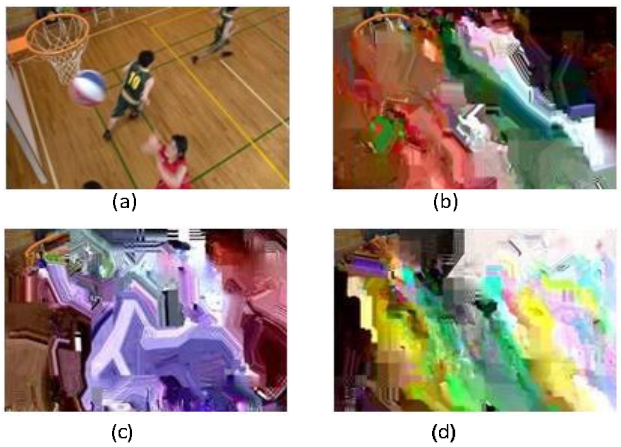


Fig.7. Original and encrypted frames decoded at QP=24: (a) original frame,(b)encrypted frame with L_{MAX}=4, (c) encrypted frame with L_{MAX}=8 and (d) encrypted frame with L_{MAX}=12.

B. Encryption space evaluation

Since the proposed technique is a selective encryption one, it is necessary to evaluate corresponding encryption space defined as the percentage of selected bits chosen for encryption from HEVC bitstream. We encrypted the 10 first frames from each benchmark sequence, with a secret

parameter L_{MAX}=14 in a two modes: firstly, in a low delay mode with each GOP composed by ones intra frame followed by 7 P-frames, and secondly in a random access mode where each GOP is composed by an intra frame followed by 7 B-frame. Table IV shows obtained results for the two tested mode. It is clear that encryption spaces for each sequence in all tested modes are very close to each other. The space's size of obtained results is less than 10% meaning that at least 10% bits of the bitstream provides sufficient high visual degradation.

The influence of QP and L_{MAX} parameters on encryption space variation is evaluated by encrypting the 10 first frame of the BasketballPass sequence. Obtained results show that encryption space increases proportionally for with L_{MAX} increasing values as shown in Table V. In contrast, the size of encryption space decreases with QP values increasing as is depicted in Table VI. Note that a reduced encryption space is obtained due to the encryption efficiency of HEVC compression.

When the proposed scheme is based upon the encryption of Golomb-Raise suffixes, it is important to outline that they have an important statistical frequency in HEVC encoded sequences. A statistical evaluation of codes employed to binarize the coeff_abs_level_remaining suffix is performed in order to evaluate the encryption space size. Table VII shows that the frequency of Golomb–Rice suffix is approximately 8 times higher than Exp-Golomb suffix for all tested sequences, which proves that the important weight of this selected code for encryption.

C. Reconstructed sequence's quality

In order to evaluate the quality of reconstructed video sequences, the simplest and most widely quality used metric is the peak signal-to-noise ratio (PSNR) expressed in decibel (dB). Table VIII lists the average PSNR for the intra decoded frames from all tested sequences encoded at QP=18 with L_{MAX}=14 in a low delay mode when using original HEVC without encryption (Original), and using the proposed selective encryption (Encrypted).

The most pertinent component that incorporates the most meaningful information is luminance component Y. Hence, it is apparent that for all tested sequences, PSNR values of luminance are less than 13dB, signifying that highest visual degradation is achieved for all tested sequences. The PSNR values of the remaining components U and V are reduced to roughly half of their original values.

The structural similarity index (SSIM) is another perceptual metric used to evaluate the quality of the proposed approach. This metric uses only luminance component for calculating (since human visual system is more sensitive to luminance than chrominance components), and it additionally evaluate the structural distortion between two frames. Table IX illustrates PSNR of luminance and the SSIM of decoded frames from BasketballPass sequence without decryption encoded at different QP values of 16, 18, 22, and 26 in a low delay mode. The results shows that for every QP value, a high visual degradation is obtained, when the PSNR values confirms a full distortion obtained using the proposed selective encryption since all values are less than 13dB. Additionally, all SSIM obtained values are less than 0.6 signifying that no visual structural correlation can be found between original and encrypted frames. Hence, the proposed selective encryption algorithm can be considered as a good encryption system with good confidentiality according to the criterions cited in [20].

A similar quality evaluation is performed with respect to the parameter L_{MAX} using several values equal to 3,4,5,6,10 and 14 encoded at QP=18 in a low delay mode. It is clear from corresponding results illustrated in Table IX that PSNR of luminance and SSIM are inversely proportional to the values of L_{MAX}, starting at 15.65dB of PSNR values, and 0.264 for SSIM ones. Best results can be achieved when encrypting the 14 possible coeff_abs_level_remaining suffixes.

D. Run-time and performances evaluation

Experimental simulations were performed on an Intel 2.3GHz Dual-Core T4500 processor with 3 Gb of memory. Table X shows timing results in millisecond (Ms) of encoding process for the first frame with and without encryption. Encoding is performed at QP=18 whereas the encryption is done using L_{MAX}=14. The difference between encoding and encryption time is negligible, and is estimated roughly as the processing time of extraction, encryption, and replacement of the plaintext before the BAC step. Encryption time can be further optimized by defining a novel implementation of HEVC encoder appropriate to the proposed algorithm.

TABLE.IV.
ENCRYPTION SPACE (IN PERCENTAGE) FOR BENCHMARKED SEQUENCES ENCRYPTED IN LOW DELAY AND RANDOM ACCESS MODE AT QP=18 WITH L_{MAX}=14

Sequences	Low delay space (%)	Random Access space (%)
BasketballPass	9.90	9.06
BasketballDrill	6.90	6.42
Johnny	7.81	6.43
BasketballDrive	3,41	2,28
Traffic	4,01	3,58
YachtRide	5,05	4,78

E. Encryption key space and plaintext's sensitivity

The proposed selective encryption algorithm uses 128 bits for the sub-key K_1, 128 bits for the sub-key K_2, and 16 to represent the value of the parameter L_{MAX} that is ranging from 1 to 14. The key space then contains $2^{128+128+4}$ possible key. Therefore, we consider that the key space is sufficiently large to permit robustness against exhaustive key search.

In order to evaluate sensitivity of the approach to plaintext variations, we encrypted the first frame of the "Johnny" sequence at QP=18 using L_{MAX}=14, K_1= A23412841234BFFF and K_2=5E198FE4128825AF then we encode in a low delay mode. The plaintext recuperated before encryption is submitted to a bit change in different places (begin, middle, and end) then encrypted again. Fig. 8 shows that for every bit change in the plaintext, decoded frame keeps a high degradation of visual content due to the randomness of

TABLE.V.
ENCRYPTION SPACE VARIATION OF FIRST ENCRYPTED FRAME FORM BASKETBALLPASS SEQUENCE ACCORDING TO QP VALUES

QP value	Encryption space (%)
18	9,90
20	9,12
24	8,22
30	4,10

TABLE.VI.
ENCRYPTION SPACE VARIATION OF FIRST ENCRYPTED FRAME OF BASKETBALLPASS SEQUENCE ACCORDING TO L_{MAX} PARAMETER'S VALUES

L_{max} value	Encryption space (%)
4	2,14
6	4,12
8	6,16
12	8,01

ciphertext resulting by encrypting all plaintexts in CBC-AES mode, which proves the robustness of the proposed algorithm.

F. Comparative study

TABLE.VII.
EXPERIMENTAL FREQUENCIES OF GOLOMB–RICE CODE AND EXP-GOLOMB CODE EMPLOYED FOR COEFF_ABS_LEVEL_REMAINING BINARIZATION IN THE HEVC ENCODING STANDARD

Sequences	Golomb–Rice code (%)	Exp-Golomb code (%)
BasketballPass	87,84	12,15
BasketballDrill	87,42	12,57
Johnny	87,12	12,87
BasketballDrive	87,41	12,58
Traffic	87,36	12,63
YachtRide	87,05	12,94

A comparative study is performed with some recent works on selective encryption of last video standards (H264/AVC, HEVC) proposed after 2010. A set of different criterions is used to evaluate and compare tested encryption algorithms. In Table 11, algorithms chosen for comparison are conform to the format of encrypted video standard, but they differ in several aspects like maintenance of compression rate, encryption domain, context modeling, encryption algorithm, and compression independence. While the scheme proposed in [12] selects coeff_abs_level_remaining suffixes and signs of transform coefficients to secure HEVC videos, and hence modify the context modeling used for BAC compression, the proposed scheme reduces the encryption space by encrypting only coeff_abs_level_remaining suffixes binarized by

TABLE.VIII.
PSNR OF DECODED FRAMES WITHOUT DECRYPTION (ORIGINAL) ,AND USING PROPOSED SELECTIVE ENCRYPTION (ENCRYPTED) IN LOW DELAY MODE AT QP=18 WITH L_{MAX}=14

	PSNR Y (dB)		PSNR U (dB)		PSNR V(dB)	
Sequence	Orig.	Enc.	Orig.	Enc.	Orig.	Enc.
BasketballPass	45.77	8.64	47.23	26.05	47.46	22.21
BasketballDrill	45.19	11.9	45.99	16.80	47.10	18.47
Johnny	46.26	9.33	49.44	21.09	49.87	23.29
BasketballDrive	46.45	9.93	46.35	9.21	48.16	13.76
Traffic	46.25	6.89	45.52	14.66	47.04	15.45
YachtRide	47.59	8.80	48.92	12.11	48.04	10.27

TABLE.IX.
VARIATION OF PSNR AND SSIM FOR FIRST DECODE FRAME OF BASKETBALLPASS SEQUENCE ACCORDING TO L_{MAX} AND QP VALUE

L_{max} Evaluation			QP Evaluation		
Parameter Value	PSNR Y (dB)	SSIM	Parameter Value	PSNR Y (dB)	SSIM
3	15.65	0.264	16	10.89	0.222
4	12.33	0.287	18	8.64	0.020
5	13.03	0.286	22	9.19	0.164
10	11.77	0.148	26	11.31	0.130
14	8.64	0.020	28	10.12	0.110

TABLE.X.
ENCRYPTION TIME (MS) ESTIMATED FOR FIRST FRAME ENCODING OF EACH BENCHMARK SEQUENCE WITH AND WITHOUT ENCRYPTION (QP=18)

Sequences	Encoding Time (Ms)	
	Without Encryption	With Encryption
BasketballPass	22.94	23.08
BasketballDrill	90.81	90.90
Johnny	162.62	162.69
BasketballDrive	444.15	444.23
Traffic	845.12	845.94
YachtRide	1520.10	1521.01

Golomb–Rice code without any change that affect the context modeling of BAC compression. This result in optimized encryption and enhanced format complacence with extremely effective encryption performances.

V. CONCLUSIONS

Emerging high Efficiency Video Coding standard HEVC presents new compression concepts such as Golomb–Rice codes that can be considered as good support to ensure security of selective encryption. We have presented in this paper a novel scheme of selective encryption based on the protection of Golomb–Rice suffixes (*coeff_abs_level_remaining*) using AES-CBC enciphering algorithm. We selected only suffixes of sub-blocks belonging to intra slice (I_SLICE), and the encryption is performed before binary arithmetic coding (BAC).

According to obtained results, we show that visual content of decoded frames from encrypted bitstream is very low for all video resolutions, implying that high visual degradation is attained using the proposed scheme. Decoding without errors confirms the format compliance of the encrypted bitstream, while the proposed approach permitted to obtain a reduced

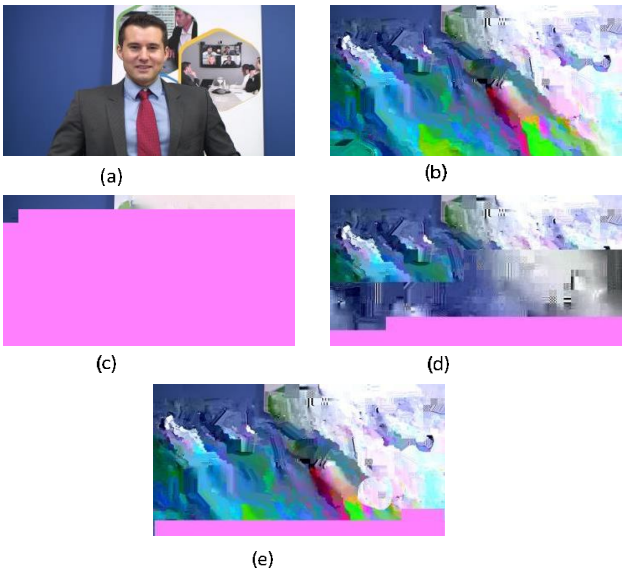


Fig.8. Decoded frame results for plaintext sensitivity evaluation: (a)original frame, (b) encrypted frame, (c)bit changed at beginning of plaintext, (d)bit changed at middle of plaintext, and (e) bit changed at end of plaintext.

encryption space formed by a minimal set of encrypted bits. We also compare the processing time of encoding with and without encryption, and we show that encryption overhead is negligible difference with respect to encoding one, implying that the scheme is suitable for real time applications.

To measure the distortion between original and encrypted frames, we utilized PSNR and SSIM metrics. Experimental results justify the high visual degradation obtained for all QP and L_{MAX} used values. Furthermore, the plaintext sensitivity is benchmarked against bit change in several locations of the encrypted sequence, and decoded results show that the proposed scheme provides high sensitivity.

We rewind that the scheme depends on compression quality of HEVC. Thus, better results can be achieved for low QP values (less than 24) since the size of the encryption space is inversely proportional to QP values. We note that the proposed technique is one of the first selective encryption techniques characterized by format compliance and optimized encryption space for the HEVC encoding standard.

TABLE.XI.
COMPARATIVE ANALYSIS BETWEEN OUR APPROACHES AND PRIOR SELECTIVE ENCRYPTION ALGORITHMS

Encryption algorithm	Compression ratio maintained	Encryption domain	Context modeling	Encryption algorithm	Compression independence
Wei et al (H.264/AVC)[21]	No	NALUs	No	RC4 + XOR	Yes
O.-Y. Lui (H.264/AVC) [9]	Yes	DCT coefficient (Sign of T1, Non-zero level)	No	Chaos + XOR	Yes
Shahid et al. (H.264/AVC) [8]	Yes	DCT coefficient (Sign of T1, Non-zero level)	No	AES+XOR	Yes
Shahid et al. (HEVC) [12]	Yes	coeff_abs_level_remaining suffix + signs	YES	AES+XOR	Yes
Wang et al. (H.264/AVC) [14]	Yes	DCT coefficient (Macro-blocks)	No	Permutation	No
Proposed algorithm	Yes	coeff_abs_level_remaining suffix	No	AES-CBC	Yes

V. REFERENCES

[1] Sullivan, G. J., Ohm, J., Han, W. J., & Wiegand, T. (2012). Overview of the high efficiency video coding (HEVC) standard. Circuits and Systems for Video Technology, IEEE Transactions on, 22(12), 1649-1668.

[2] Wiegand, T., Sullivan, G. J., Bjontegaard, G., & Luthra, A. (2003). Overview of the H. 264/AVC video coding standard. Circuits and Systems for Video Technology, IEEE Transactions on, 13(7), 560-576.

[3] ISO/IEC JTC1/SC29/WG11 Vision, Application, and Requirements for High Performance Video Coding (HVC), MPEG Document, N11096, Kyoto, JP (Jan. 2010).

[4] Sze, V., & Budagavi, M. (2012). High throughput CABAC entropy coding in HEVC. Circuits and Systems for Video Technology, IEEE Transactions on,22(12), 1778-1791.

[5] Marpe, D., Schwarz, H., & Wiegand, T. (2003). Context-based adaptive binary arithmetic coding in the H. 264/AVC video compression standard. Circuits and Systems for Video Technology, IEEE Transactions on, 13(7), 620-636.

[6] High efficiency video coding, ITU-TRec.H.265 and ISO/IEC 23008-2 (MPEG-H, Part 2), Apr.(2013), version 1.

[7] Sole, J., Joshi, R., Nguyen, N., Ji, T., Karczewicz, M., Clare, G., ... & Duenas, A. (2012). Transform coefficient coding in HEVC. Circuits and Systems for Video Technology, IEEE Transactions on, 22(12), 1765-1777.

[8] Shahid, Z., Chaumont, M., & Puech, W. (2011). Fast Protection of H. 264/AVC by Selective Encryption of CAVLC and CABAC for I and P frames. Circuits and Systems for Video Technology, IEEE Transactions on, 21(5), 565-576.

[9] Lui, O. Y., & Wong, K. W. (2013). Chaos-based selective encryption for H. 264/AVC. Journal of Systems and Software, 86(12), 3183-3192.

[10] Park, S. W., & Shin, S. U. (2008, September). Efficient selective encryption scheme for the H. 264/scalable video coding (SVC). In Networked Computing and Advanced Information Management, 2008. NCM'08. Fourth International Conference on (Vol. 1, pp. 371-376). IEEE.

[11] Wang, X., Zheng, N., & Tian, L. (2010). Hash key-based video encryption scheme for H. 264/AVC. Signal Processing: Image Communication, 25(6), 427-437.

[12] Shahid, Z. Puech, W., Visual Protection of HEVC Video by Selective Encryption of CABAC Binstrings, Multimedia, IEEE Transactions on , vol.16, no.1, pp.24,36, Jan. 2014
[13] Van Wallendael, G., Boho, A., De Cock, J., Munteanu, A., & Van de Walle, R. (2013, January). Encryption for High Efficiency Video Coding with video adaptation capabilities. In Consumer Electronics (ICCE), 2013 IEEE International Conference on (pp. 31-32). IEEE.
[14] Wang, Q., & Wang, X. (2014, May). A new selective video encryption algorithm for the H. 264 standard. In Progress in Informatics and Computing (PIC), 2014 International Conference on (pp. 275-279). IEEE.
[15] Zhang, X., & Qiu, B. (2014). Fast Mode Decision and Encryption Policy in H. 264/AVC Frame-skipping Transcoding. Journal of Computers, 9(5), 1201-1208.
[16] Nithya, B., & Radharani, S. (2014). Scanned Document Compression Using High Efficiency Video Coding (HEVC) Standard.International Journal, 3(11).
[17] Zhou, J., Liu, X., Au, O. C., & Tang, Y. Y. (2014). Designing an efficient image encryption-then-compression system via prediction error clustering and random permutation.
[18] Dherbecourt, Y. M., Herodin, J. M., & Vidrascu, A. (1996). U.S. Patent No. 5,583,940. Washington, DC: U.S. Patent and Trademark Office.
[19] SHM reference software: https://hevc.hhi.fraunhofer.-de/svn/svn_SHVCSoftware/
[20] Dubois, L., Puech, W., Blanc-Talon, J., 2012. Reduced selective encryption of intra and inter frames of H.264/AVC using psychovisual metrics. In: 19th IEEE Inter-national Conference on Image Processing, pp. 2641–2644.
[21] Wei, Z., Wu, Y., Ding, X., & Deng, R. H. (2012). A scalable and format-compliant encryption scheme for H. 264/SVC bitstreams. Signal Processing: Image Communication, 27(9), 1011-1024.

Mokhtar Ouamri received his engineer degree from the University of Sciences and Technologies of Oran (USTO), Oran, Algeria, in 2007, and the M.S. degree from the University of Sciences and Technologies of Oran (USTO), Oran, Algeria, in 2010, both in computer science. Since December 2011, he is PhD candidate in computer science, at Djillali Liabes University (UDL), Algeria. He is currently Assistant Professor at University of Ibn-Khaldun, Tiaret, Algeria. His research interests are in the fields of multimedia compression/security, image/video processing and analysis, video surveillance (detection, tracking, event detection, and storage) ,and multimedia communication.

K.M. Faraoun was born in Sidi Bel abbes, Algeria, in February 23, 1978. He received his master's degree in computer science at the computer science department of Djilali Liabbes University- Sidi-Bel-abbes – Algeria in 2002, his Ph.D degree in computer science, in 2006, and his HDR degree in computer science and intelligent systems, in 2009 From UDL-University. His current research areas include computer security systems; cryptography; genetic algorithms; cellular automata; evolutionary programming and information theory. Dr. Faraoun is currently a Full professor and a teacher at the computer sciences department of UDL-University, he teaches Information Theory and Cryptography. He has published several papers in many international journals.

J Inf Process Syst,
http://dx.doi.org/10.3745/JIPS

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

New Compliant Scheme for Selective Encryption of HEVC/H.265 Videos

Mokhtar Ouamri*, Kamel Mohamed FARAOUN **

Abstract

The aim of this paper is to propose a novel scheme of selective encryption (SE) for High efficiency video coding standard (HEVC) according to *its latest Draft International Standard*. The main contribution of this work is to define for the first time a sufficient encryption space of encrypt-able quantized transform coefficients QTCs by preserving the encoding efficiency of encrypted bitstream (format compliancy , and same bit rate) during CABAC (Context-adaptive binary arithmetic coding) entropy coder compared with previous SE-HEVC schema. The latest standardized draft offers us helpful solutions in designing an efficient cryptosystem especially in QTCs binarization. Bins of suffixes as well as signs of selected QTCs are packed together to form plaintexts that are ciphered by advanced encryption systems (AES) in CBC mode.
Experimental results show efficiency and robustness of the proposed technique approved by several security tests. Moreover, a high visual degradation is obtained in low, medium, and high quality of compression's level assessed by visual metrics with a considerable encryption space. We also show by comparative analysis with previous SE-HEVC that the proposed approach is one of the first cryptosystem that generate a decodable enciphered bitstream according to the latest standardized draft of HEVC. In addition, our approach does not require updating the context modeling and it will also help us in reducing the complexity in time processing as well.

Keywords

High efficiency video coding, selective encryption, Context-adaptive binary arithmetic coding, advanced encryption system

1. Introduction

With the advent of its first standard draft in April 2013 [1], the milestone high efficiency video coding standard (HEVC) [2] will dominate all multimedia research fields for its efficiency compared to its predecessor H.264/AVC [3]. Digital right management (DRM) will certainly take a fundamental place for its importance to ensure the protection with secure distribution of HEVC video content.

Selective encryption (SE) is one of the ideal solutions to achieve the DRM requirements for multimedia content security. It consists of encrypting only selected subset of multimedia data, and it provides typically significant results when is performed jointly during the compression process. To preserve HEVC compression efficiency, SE should be carried out during the entropy coding module.

※ This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Manuscript received: December 23, 2014; accepted July 28, 2015.
Corresponding Author: Mokhtar Ouamri (amokhtar124@yahoo.fr)
* The Department of Computer Science,Djillali Liabes University of Sidi Bel Abbes, Sidi Bel Abbes 22000, Algeria(amokhtar124@yahoo.fr)
** The Department of Computer Science,Djillali Liabes University of Sidi Bel Abbes, Sidi Bel Abbes 22000, Algeria(kamel_mh@yahoo.fr)

Copyright© XXXX KIPS

Context-adaptive binary arithmetic coding (CABAC) [4] is the single lossless entropy coder employed in HEVC. Its global schema is identified by three main operations: binarization, context modeling, and binary arithmetic coding (BAC), and with two operating modes: regular mode which requires predefined context models for coding, and bypass mode which assumes 0.5 of probability for each symbol.

Before the standardization of HEVC in April 2013, transform coding has occupied extensively considerable scope in the research area. Firstly, a family of truncated rice codes (called also Golomb-rice codes) [5] is newly introduced for levels' binarization, which does not require context modeling; secondly, levels up to predefined thresholds should be binarized with truncated rice code, while the remaining levels should be binarized with Exp-Golomb code [6].

In the old working drafts like WD6 [7], one of the first attempts on levels binarization was adopted [8]. It was distinguished by thresholds of transitions between two families of codes [7, Table 9.35], [9], and zeroth order Exp-Golomb coding. Shahid *et al.* proposed in two works [10]-[11] the protection of both levels and signs of quantized transform coefficients (*QTC*s) according to WD6. In their first work [10], non-dyadic (i.e., not-encrypt-able) truncated rice codes were decomposed into small dyadic codes followed by a change in context modeling. In their second work [11], they adapted their prior work of SE-H.264 [12] for protecting both binary codes of truncated rice and Exp-Golomb.

Transform coding in the latest standard draft of HEVC [1], [13] is changed entirely in several aspects mostly in levels binarization where the old scheme of levels binarization was substituted by new another one [14]. Consequently, the entire solutions proposed in [10]-[11] become not appropriate, and the novel HEVC decoder will not be able to read both plain and encrypted bitstream owing to the adopted draft WD6 [7]. Moreover, we cannot retrieve theirs ciphertexts during decoding process owing to the profound modification done in transform coding structure; accordingly, we conclude that the challenge of *QTC*s encryption during the entropy module should be resolved again. The main contribution of this paper is to overcome this challenge by proposing a novel SE of *QTCs* in accordance with the draft international standard of HEVC [1]. We show that this draft provides us helpful solutions for designing an efficient SE unlike the old working drafts. We define for the first time a sufficient set of encrypt-able *QTCs* that permits us to protect HEVC bitstream in CABAC mode without altering the encoding performances of HEVC standard. Both *QTCs* levels and signs will be encrypted in compliant manner with AES in CBC mode. The protected bitstream will preserve the same bit rate without any change (increase/decrease), and the decoding results of the proposed approach provide us high visual degradation with sufficient confidentiality.

The remaining of this paper is organized as follows: a brief description of the HEVC structure with corresponding coding tools, transform coding according to the latest HEVC standard draft and prior encryption works are reviewed in Section 2. Section 3 describes the proposed SE scheme with its choice of encryption space, and section 4 outlines the obtained results of the performed experiments. Finally, conclusions are drawn in section 5.

2. Background and related works

2.1. Description of HEVC structure and coding tools

The emerging high efficiency video coding (HEVC) is a hybrid video codec, which employs hybrid spatial-temporal prediction schema [2, Fig. 1] as H.264/AVC for compressing a raw video into a binary stream (bitstream) by minimizing a computational rate-distortion (R-D) cost.

The optimization of R-D cost is attained by the block partitioning structure [2] identified by three basic units: coding unit (CU), prediction unit (PU), and transform unit (TU). The latter is a leaf of a nested tree structure resulting from the transformation of residual signal through the residual quad-tree transform (RQT) [15].

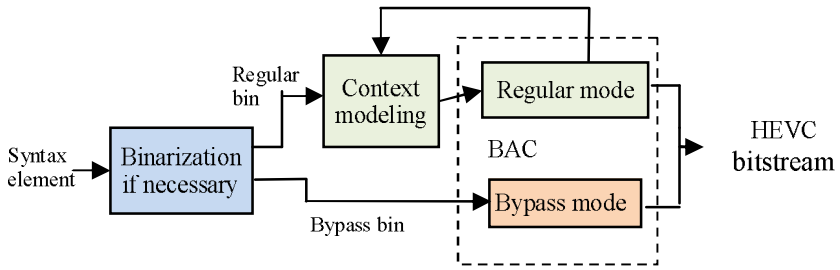


Fig. 1. Bloc diagram of CABAC engine with three key operations: binarization, context modeling, and binary arithmetic coding (BAC).

CABAC (Context adaptive binary arithmetic coding) [4] is the single entropy coder in HEVC standard. It involves three main operations depicted in Fig. 1: binarization to decompose non-binary syntax elements into a sequence of binary decisions (so-called bins), context modeling, and binary arithmetic coding (BAC). Each binary symbol is coded either by regular mode where the context models are required for bins encoding or by bypass mode where bins are coded with equi-probability.

Five basic binarization codes [13] are used, including: unary coding, truncated unary coding, *k*th order Exp-Golomb (EG_k) coding, truncated rice coding of order k (TR_k) and fixed length coding (FL). Given an unsigned integer x, the unary code consists of a number x of "1" bits plus a terminating "0" bit. For truncated unary code, the unary code is used only when $x<cMax$ ($cMax$=4 as is defined in [1]). If $x=cMax$, the terminating "0" bit is neglected. The FL code of x is just x with a fixed number $\ell=\lceil \log_2(cMax+1) \rceil$ of bits.

A truncated rice code (known also as Golomb-rice code) [5] is an optimal code for representing a symbol value x and is defined as the quotient $q = [x/2^k]$ and the remainder $p = x - q \times 2^k$ whereas k is a rice parameter. The quotient represents the prefix part binarized with a unary code, and the remainder represents the suffix part composed of k length bins. The Exp-Golomb code [6] of symbol value x is obtained by concatenation of the prefix and suffix codewords. The prefix is the unary code

of $\ell(x) = \log_2(\frac{x}{2^k} + 1)$, whereas the suffix is calculated by $x+2^k(1-2^{l(x)})$

Many techniques were added to improve the throughput [16], including reducing context coded bins, grouping bypass bins together, grouping bins that use the same contexts together, reducing context selection dependencies, and reducing the total number of signaled bins. Furthermore, the HEVC defines a new set of entropy syntax elements for describing the properties of transform coefficients coding

2.2. Brief review of transform unit structure

Fig. 2. Transform units TUs ranging from 4×4 to 32×32 of encoded foreman sequence in low delay mode at QP=18

```
/*Last Significant Coefficient Coordinates*/
last_significant_coeff_x_prefix;
last_significant_coeff_y_prefix;
if( last_significant_coeff_x_prefix > 3)
last_significant_coeff_x_suffix;
if( last_significant_coeff_y_prefix > 3)
 last_significant_coeff_y_suffix;
```

```
for(i = numLastSubset; i >= 0; i--)
{
                    /*Significance Map*/
 coded_sub_block_flag[i];
for(n = 15; n >= 0; n--)
{
if(coded_sub_block_flag)
 significant_coeff_flag[n];
}
          /* QTCs coding for each 4×4 sub-block*/
for(n = 15; n >= 0; n--)
{
if(significant_coeff_flag[n])
 coeff_abs_level_greater1_flag[n];
if(coeff_abs_level_greater1_flag[n])
 coeff_abs_level_greater2_flag[n];
if(significant_coeff_flag[n])
 coeff_sign_flag[n];
if(coeff_abs_level_greater2_flag[n])
 coeff_abs_level_remaining[n];
}
}
```

Fig. 3. Pseudo code of encoding of syntax elements related to a transform unit

RQT transform decomposes each CU into transform units TUs of sizes varying from 4×4 to 32×32. Fig. 2 shows an example of the decoded frame of foreman sequence (in CIF format) and corresponding TUs.

The related blocks TBs (luminance and chrominance) of TUs are encoded as is illustrated by the

pseudo code of Fig. 3 [13], [14]. First, each non-zero QTC is considered as significant coefficient, and the position of the last significant coefficient of TB is firstly encoded; later, TBs larger than 4×4 QTCs are decomposed into 4×4 sub-blocks, and for each sub-block, coded_sub_block_flag is used to indicate whether the sub-block is empty (all QTCs are zero) or not.

QTCs within each non-empty sub-block are encoded in the inverse diagonal scan order, and five syntax elements are signaled to represent the *QTCs* levels: *significant_coeff_flag*, *coeff_abs_level_greater1_flag*, *coeff_abs_level_greater2_flag*, *coeff_sign_flag*, and *coeff_abs_level_remaining*. Table 1 describes the semantic of each syntax element. To achieve more compression efficiency, all *significant_coeff_flag* syntax elements of each non-empty sub-block will define the significance map, and they will be encoded together. Likewise, the four remaining syntax elements are grouped together. Firstly, the first two groups of syntax elements will be encoded in regular mode, later the last two groups will be encoded in bypass mode.

Table 1. Description of CABAC syntax elements employed for level coding

Syntax element	Description
significant_coeff_flag	indicates the significance of each coefficient
coeff_abs_level_greater1_flag	indicates whether the coefficient amplitude is larger than one for each non zero coefficient
coeff_abs_level_greater2_flag	indicates whether the coefficient amplitude is larger than two for each coefficient with amplitude larger than one
coeff_sign_flag	indicates sign information of the nonzero coefficients
coeff_abs_level_remaining	indicates remaining absolute value of coefficient level

2.3. Binarization and transform coding

Since *significant_coeff_flag*, *coeff_abs_level_greater1_flag*, *coeff_abs_level_greater2_flag*, and coeff_*sign_flag* are binary symbols, *coeff_abs_level_remaining* is the unique syntax element that needs to be binarized (transformed into binary decisions i.e., bins) before coding with BAC engine.

The *coeff_abs_level_remaining* value [1, Sec. 9.3.3.9], [14] is computed by following equations:

$$coeff\_abs\_level\_remaining = |QTC| - baselevel \quad (1)$$

when *baselevel* can be 1,2,or 3 and it is calculated on the basis of previous *QTCs*.

Given rice parameter *p*, *coeff_abs_level_remaining* is binarized by TR_p if its value is inferior to threshold $TR_{max}[p]$, otherwise *coeff_abs_level_remaining* - $TR_{max}[p]$ is binarized using EG_{p+1} prefixed by 1111 as illustrated in Fig. 4. Threshold $TR_{max}[p]$ depends on rice parameter *p* in following manner:

$$TR_{max}[p] = 4 \times 2^p \quad , p \in \{0,1,2,3,4\} \ . \quad (2)$$

thus $TR_{max}[p] \in \{4,8,16,32,64\}$(e.g., $TR_{max}[p]=16$ for $p=2$).

The Rice parameter is updated later as following:

$$p_{next} \leftarrow min\,(p+1,4), if\ |QTC| > 3 \times 2^p \quad (3)$$

Table 2 gives us an example of *QTCs* codes when $p=2$.

BAC is an efficient entropy coding method, where a finite sequence of bins is represented by an offset

inside an interval (range) (HEVC uses 9 bits for offset coding and 10 bits for range coding). *significant_coeff_flag*, *coeff_abs_level_ greater1_flag*, and *coeff_abs_level_greater2_flag* are encoded in regular mode, which uses adaptive probability models (look up tables for context models). However, *coeff_abs_level_remaining* and *coeff_sign_flag* are encoded in the low-complexity bypass mode by assuming equal probability of 0.5 for both bins values.

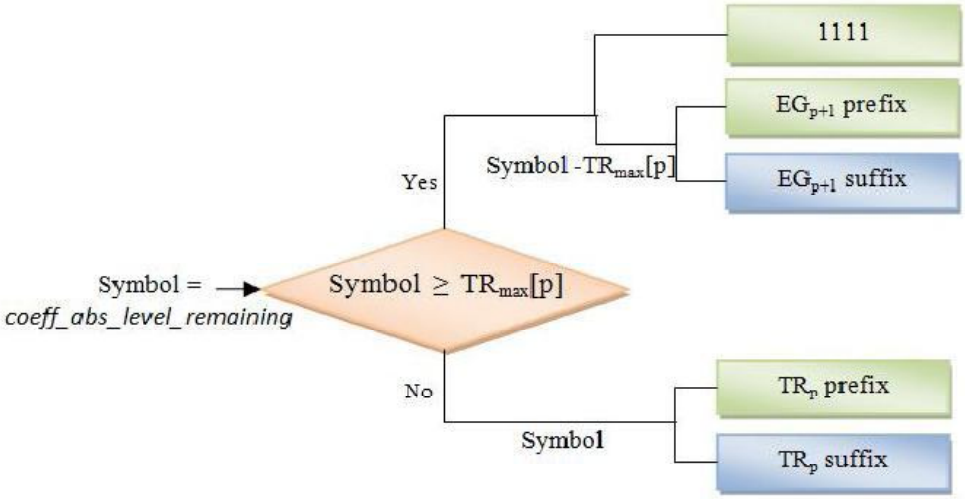


Fig. 4. Schema of binarization of QTCs levels according to HEVC standard draft [1].

2.4. Related works

The state of the art of selective encryption approaches is rich for video standards. Most of them were designed for H.264 standard due to its emergence in the last decade. Among them, [17]-[20] and [12] focused on preserving the bitstream length by encrypting entropy syntax elements in complaint manner. H.264 uses either CAVLC [32] or CABAC for entropy coding. [17] and [18] proposed SE-CAVLC to protect H.264 bitstream while [19]-[21] investigated CABAC structure and found sufficient data to be encrypted in compliant manner.

HEVC improves the CABAC throughput by coding a large set of syntax elements in bypass mode, which constitutes a good help for encryption since such as data do not depend on context models.

In [21], Wallendael *et al.* studied in depth the impact of encryption of some CABAC syntax elements on bit rate and visual quality. The encrypted data can belong to CU (e.g., delta QP), to PU (e.g., motion vector difference (MVD)), or to transform unit like QTCs' signs. They found that only bypass syntax elements could retain the bit rate of encrypted bitstream; moreover, they found that the protection of QTCs' signs and MVD signs enhances significantly the perceptual content security of encrypted video.

Hofbauer *et al.* proposed in [22] a transparent encryption scheme by encrypting bits of signs of *ACs* coefficients over a wide range of quantization parameters. The principal aims are the format compliancy, length preservation of bitstream, and low visual quality. The latter is the main purpose of transparent encryption.

In [23], we presented a previous approach of SE-HEVC by encrypting a set of suffixes of TR_p codes. We found that it provides sufficient results in terms of visual security and time processing. Unfortunately, this approach works well only for low delay mode.

In [24], V. A. Memos *et al.* presented in their work a solution of SE-HEVC by protecting a set of DC and ACs coefficients with strong cipher. Unfortunately, this approach does not provide the same size of encrypted bitstream.

In [10], Shahid *et al.* proposed one of the first works on SE-HEVC based on *QTCs* encryption. Their work adopts one of the first realization of HEVC level coding [7]-[9] where *QTCs* were binarized using TR_p for short symbols and zeroth Exp-Golomb for long symbols [10, Fig. 2]. They proposed a procedure of conversion of non dyadic codes into small dyadic codes followed by an update on context modeling; later, they prepared a plaintext formed by dyadic codes of *QTCs* and levels sign and encrypted them using AES-CFB mode. In [11], Shahid *et al.* adapted their prior work on SE-H.264 [12, Fig. 6] for encrypting both TR_p and EG_0 codes of *QTCs* and signs levels [11, Figure 3].

Table 2. Binarization codes of QTCs when p=2, left table is for TR_p codes, right table is for EG_3 codes (not encrypt-able codes are colored in gray)

symbol	prefix	suffix
0	0	00
1	0	01
2	0	10
3	0	11
4	10	00
5	10	01
6	10	10
7	10	11
8	110	00
9	110	01
10	110	10
11	110	11
12	1110	00
13	1110	01
14	1110	10
15	1110	11

symbol	prefix	suffix
16	11110	000
17	11110	001
18	11110	010
19	11110	011
20	11110	100
21	11110	101
22	11110	110
23	11110	111
24	111110	0000
25	111110	0001
26	111110	0010
27	111110	0011
28	111110	0100
29	111110	0101
30	111110	0110
31	111110	0111

Despite the significant results of Shahid *et al.*, these solutions become not applicable, as theses ones depend on old transform coding which is changed wholly in several aspects especially on *QTCs* binarization [14].

In the next section; firstly, we will discuss the inapplicability of [10]-[11] according to the latest novelties of standard draft; later, we will propose our selective encryption approach in detail.

3. Proposed approach

The protection of *QTCs* using AES during CABAC compression is an interesting research topic, since the encryption of QTCs provides high visual degradation, and AES is always one of robust cipher owing to its high security in both software and hardware implementation. Furthermore, SE-CABAC permits to obtain a compliant encrypted bitstream with null bit rate increase.

Table 3. Binarization differences between standard draft and old working draft WD6 [10] of HEVC.

	WD6	**HEVC draft standard**
Coded value	$\|QTC-3\|$	$\|QTC\|-baselevel$
Rice parameter p	{0,1,2,3,4}	{0,1,2,3,4}
$TR_{max}[p]$	{7,14,26,46,78}	{4,8,16,32,64}
Coded value< $TR_{max}[p]$	TR_p code	TR_p code
Coded value≥$TR_{max}[p]$	Zeroth order Exp-Golomb code	$(p+1)$th order Exp-Golomb code
Rice parameter p updating	$p_{next}[\|QTC\|]=\{3,5,12,24,\infty\}$	$p_{next} \leftarrow min\,(p+1,4)$,if $\|QTC\|>3\times 2^p$

Among the five syntax elements used for representing each 4×4 sub-block's *QTCs* of TU, only *coeff_sign_flag* and *coeff_abs_level_remaining* can be encrypted since they are coded in bypass mode operation by BAC engine which does not require look up tables of context models, unlike the three remaining syntax elements where BAC coding depends on look up tables and any possible modification will cause error when decoding. Table 3 summarizes the main differences on *QTCs* coding between the HEVC standard draft [1] and the old working draft WD6 [7]; firstly, the coded value in WD6 is $|QTC|-3$, while in [3] is $|QTC|-baselevel$; secondly, threshold values TR_{max} used in [1] and WD6 are very different. In [1], TR_{max} values are multiple of four, and coded values superior to TR_{max} are binarized with EG_{p+1} while in WD6 they are binarized with EG_0; finally, the update rule of rice parameter is quite changed in [1].

Before exposing our approach and introducing our study, it is necessary to test the applicability of prior solutions of SE-QTC proposed in [10]-[11]. In the case of [10], It is clear that we cannot decompose any TR_p code into small TR_p codes because here, we will obtain additional non-zero QTCs without signs. In addition, the number of significant coefficients is signaled and fixed in the significant map of the corresponding TU, which will cause certainly errors when decoding. More precisely, if we consider that the 4×4 QTCs of transform block are binarized with TR_p coding, and if we assume that all TR_p codes (i.e., 4×4 QTCs' levels) are non-dyadic, then any decomposition of them means merely that the decoder will find more than 4×4 TR_p codes for the same transform block when decoding!; furthermore, the maximum length of TR_p suffixes in the standard draft is 4, and we cannot find suffix of length equal to 14 [10, Sec. IV-B]. As a result, SE approach proposed in [10] which consists of decomposing of non dyadic codes into small dyadic codes becomes not applicable in any way. Similarly, we cannot apply the solution proposed in [11] since suffixes of both TR_p and EG_{p+1} codes in new transform coding [1] differ in format from ones employed in this approach [11, Figure 3]. In addition, EG0 code is omitted from new standard draft for QTCs binarization.

In order to obtain compliant encrypted bitstream with the same bit rate after encryption of *QTCs*, the following conditions should be respected:

1) We can encrypt only TR_p codes when $p\geq 1$ because TR_0 codes have a fixed format defined as a sequence of "1" terminating by "0."
2) Since levels are compressed by BAC engine in bypass operation mode, the rice parameter p should remain unchanged after encryption because it indicates the number of bins sequence to be decoded; in fact, any change affect p will modify the number of decoded bins;

consequently, it could cause bit rate change, and probably it will provoke decoding errors.

3) The format of code (TR_p, EG_{p+1}) should be respected.

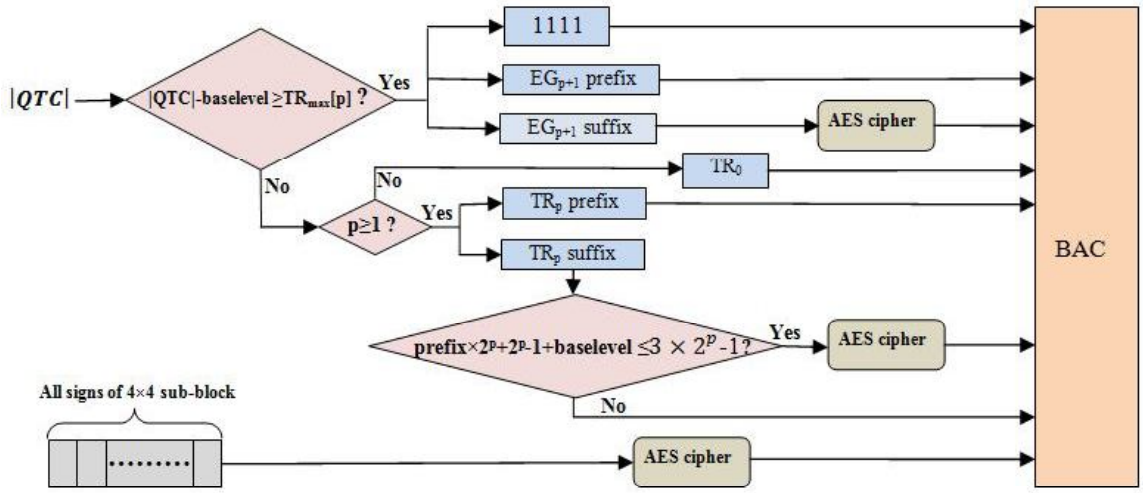


Fig. 5. Flowchart of the proposed approach illustrates the process of encryption of both signs and levels.

The HEVC draft standard offers helpful solutions for SE-CABAC of QTCs, firstly; it offers us an encryption space of 3×2^p encrypt-able TR_p elements; secondly, threshold TR_{max} is multiple of 4 which helps us while encrypting TR_p codes. Table 2 presents the binary codes of QTCs for p=2. Each symbol is composed of a prefix followed by a suffix. Since p=2, all symbols less than $TR_{max}[2]$=16 are binarized with TR_p codes and theirs suffixes consists of p bins. Each pattern of prefix has all combinations of possible suffixes. For example, when prefix=10, suffix can be 00, 01, 10, and 11. Symbol superior to $TR_{max}[2]$ are binarized by concatenation of 1111 and EG_{2+1}. Consequently, each prefix is composed of 1111 followed by EG_{p+1} prefix. In contrast to TR_p suffixes, EG_{p+1} suffix starts with $p+1$ bins and these number increments geometrically. The properties discussed for p=2 are the same for remaining p values. We conclude that SE can be performed only for suffixes bins of $QTCs$.

First of all, we note that the relation 3×2^p-1<$TR_{max}[p]$ should be always verified; furthermore, EG_0 is totally discarded from the new standard. To respect the updating rule given by equation (3), three possible cases can be cited:

1) $|QTC| \leq 3\times2^p$-1 and p≥1: in this case, p stays constant after coding. Each prefix has all possible combinations of suffixes. Consequently, any encryption affect suffix's bins will result new coefficient $QTCe$ with $|QTCe| = prefix\times2^p + suffix_e + baselevel$ .To ensure that p remains constant; we must verify whether the maximum value of $|QTCe|$ is less than 3×2^p-1. This maximum can be computed by $prefix\times2^p + 2^p - 1 + baselevel$.

2) $3\times2^p - 1 < |QTC| < TR_{max}[p]$ and p≥1: p changes only for $|QTC| < 3\times2^p < TR_{max}[p]$ while it remains constant when $|QTC| = 3\times2^p$. As a result, we cannot encrypt QTC since we cannot obtain all possible suffixes for prefix of 3×2^p. For example, when p=2, both symbols 12,13,14,and 15 have the same prefix 1110, however p will be incremented only

after coding of 13,14,15 while it will remain unchanged for 12. If we attempt to substitute 12 by 13 when encrypting levels, p will be incremented after decoding and it will cause errors in decoding of encrypted bitstream.

3) $|QTC| \geq TR_{max}[p]$ and p≥0: EG_{p+1} is employed for binarization and p is incremented after coding with or without encryption. Accordingly, suffix's bins can be encrypted without any concerns.

The possible *QTCs* that can be selected for encryption are those respecting the first and the last case. In the proposed approach, the set of *QTCs* defines the encryption space and will be denoted by E in following sections. The flowchart of Fig. 5 resumes the selection and the encryption of levels' suffixes and *QTCs*' signs.

3.1. Preparing the plaintext

It is well known that the protection of syntax elements encoded in regular mode can damages the compliance of encrypted bitstream because these data should be decoded by context models defined in predefined tables when decoding. In the case of our approach, levels' suffixes and *QTCs*' signs are encoded in bypass mode. In addition, contexts initialization is invoked only at the beginning of each slice/tile.

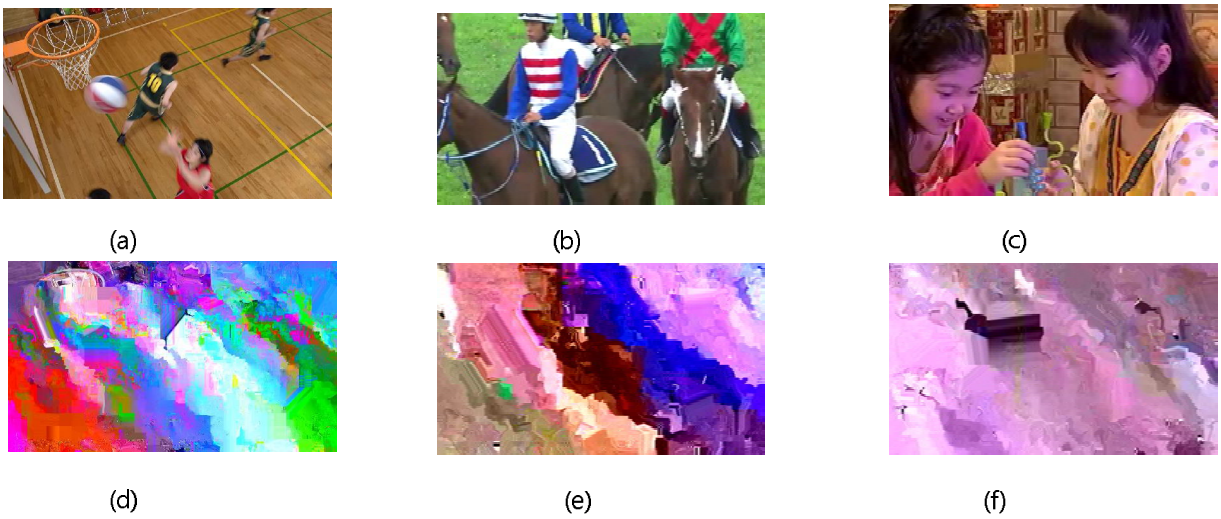
For each slice/tile, and for each corresponding color components, we prepared two plaintexts: the first contains the bins of *coeff_abs_level_remaining* suffixes concatenated together from all parsed $QTCs \in E$ found in all TUs parsed before BAC compression, while the second contains signs of all parsed non-zero *QTCs*. Finally, we encrypted later signs and levels separately to ensure high level of randomness.

Signs in HEVC are binarized by FL codes. All signs of non-zero *QTCs* within a 4×4 sub-block are encoded together in bypass mode; accordingly, for each parsed 4×4 sub-block, we append the group of signs directly into the second plaintext.

3.2. Encryption /decryption of levels and signs

Advanced encryption system (AES) [25] is one of the robust existing cryptosystems using a symmetric key of on 128,192 or 256 bits. AES is based on a design principle known as a substitution-permutation network, combination of both substitution and permutation, and is fast in both software and hardware. Each round of AES consists of several processing steps, each containing four similar, but different stages, including one that depends on the encryption key itself. A set of reverse rounds are applied to transform ciphertext back into the original plaintext using the same encryption key. AES is considered to be secure against all known cryptanalysis attacks until today except brute force attack since the latter tries all possible combinations of key to broke AES system.

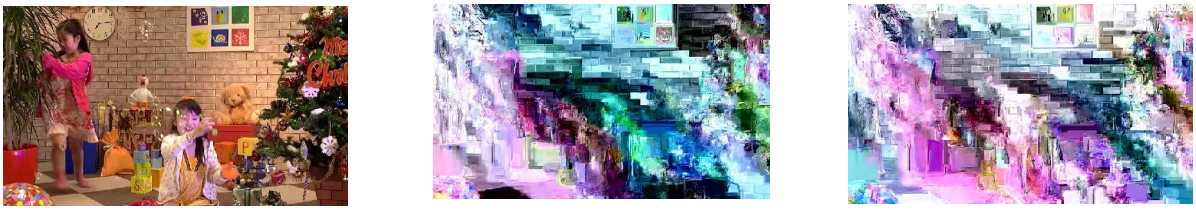
Plaintext of levels suffixes are encrypted with AES in CBC mode [26]. Practically, we achieved this by dividing the plaintext into a sequence of $n+1$ consecutive blocks $(X_1, X_2, \ldots, X_n, Y)$ where $n \geq 0$ and the length of each X_i is fixed to 128 bits, whereas the length of remainder plaintext Y is less than 128



(a) (b) (c)

(d) (e) (f)

Fig. 6. Visual results of the proposed SE decoded in Intra only mode at QP=18: (a),(b),and (c) show original videos,(d),(e),and (f) show encrypted videos



(a) (b) (c)

Fig. 7. Propagation of intra frame protection to following frames . (a) shows the original frame (intra) of PartyScene sequence ,(b) and (c) are the decoded frame at QP=28 of encrypted PartyScene sequences :(b) shows following P frame decoded in low delay mode ,and (c) shows following B frame decoded in random access mode

bits. A random initial vector $IV \in \{0,1\}^{128}$ is then chosen at random, and the ciphered block C_1 is generated by applying the AES cipher to $X_1 \oplus IV$ ($\oplus$ denotes or-exclusive bit to bit operation) using a secret key K_1 with a length of 128 bits. For $i \in \{2, \ldots, n\}$, the remainder cipher blocks C_i are obtained by encrypting $C_{i-1} \oplus X_i$ with AES using K_1. The latest block cipher C_{n+1} is a special case, and it is ciphered by $C_{n+1} = Y \oplus K_2$, when K_2 is a secret key on 127 bits that it different from K_1 to keep the privacy of AES-CBC mode. Padding [10, Sec. III-C] is not recommended because it will damage the format compliancy of the bitstream. Finally, *ciphertext* is obtained by composing the sequence $(C_1, C_2, \ldots, C_n, C_{n+1})$.

We replaced later each selected suffix's bin of plaintext by its corresponding bin in ciphertext respectively before the BAC compression. The decryption is performed after BAC decompression by the construction of the corresponding ciphertext of the suffixes' bins. Each ciphertext block is replaced by its corresponding decrypted plaintext bins.

In the case of signs encryption, the second plaintext will be encrypted firstly as the first plaintext; later, each encrypted sign will replace its plain one in the group's signs of 4×4 sub-block before transmitting

them to BAC engine.

4. Experimental results

This section presents experimental results related to the proposed approach. Since the proposed SE-HEVC cryptosystem is performed during HEVC compression, we have integrated SE module in the HEVC reference software HM v10.0 [27]. The evaluation of SE performance is carried out by using 18 YUV 420 video sequences shown in Table 4. These were selected from 18 benchmark sequences specified in the common test conditions of JCT-VC document [28] according to different visual characteristics such as texture, object, and motion.

Table 4. Benchmark sequences used for experimental results.

Class	Resolution	Frames/second	Video	Frames to be encoded
A	2560x1600	30	Traffic(S01), PeopleOnStreet(S02), Kimono,	100
B	1920x1080	24	Kimono1 (S03), ParkScene (S04)	100
C	1920x1080	50-60	Cactus (S05), BDrive (S06),BQTerrace (S07)	100
D	832x480	30-60	BasketballDrill (S08), BQMall (S09), PartyScene (S10), RaceHorses (S11)	100
E	416x240	30-60	BPass (S12), BQSquare (S13), BlowingBubbles (S14), RaceHorses (S15)	100
F	1920x1080	60	Vidyo1 (S16), Vidyo2 (S17), Vidyo3 (S18)	100

4.1. Evaluation of Encryption performance

4.1.1. Visual quality assessment

Firstly, we have encrypted 100 frames for each sequence in all tested modes. GOP in low delay mode is composed from an intra frame followed by 7 predicted (P) frames, while the one in random access mode consists of intra frame followed by bi-predicted (B) frames. Fig. 6 shows the *nin*th frame of plain sequences chosen for illustration, and the corresponding ciphered frames decoded at quantization parameter QP=18 in intra-mode only. It is apparent that the commercial values of decoded frames are fully destroyed with high visual degradation, and perceptual content is completely disguised. The impact of intra frame protection propagates as well to following frames (P and/or B frame) as is shown in Fig. 7 when we have encrypted PartyScene sequence at QP=28 in both modes of low delay and random access. Decoded results without errors confirm the format compliancy of the proposed approach.

We have evaluated the quality of reconstructed video sequences by the peak signal-to-noise ratio (PSNR) metric which is expressed in decibel (dB). Table 5 and 6 list the PSNR average [29] of the 100 decoded frames at QP=18 and at QP=32 respectively of all benchmark sequences encrypted by our

proposed SE in the three tested modes. The most pertinent component that incorporates the most significant information is the luminance component Y. Hence, it is apparent that for all tested modes, the average values PSNR of luminance are less than 9.5 dB for QP=18, which signifies that highest visual degradation is achieved in all tested sequences. The average PSNR values of the remainder components of chrominance (U and V) for all encrypted sequences are reduced to roughly half of their original values. In the same way, PSNR values remains below than 10.2 dB for luminance component which means that the perceptual content is well secured for QP=32.

Table 5. PSNR average of 100 frames per sequence encrypted in all tested modes at QP=18.

	Intra only mode			Low delay mode			Random access mode		
	Y	Cb	Cr	Y	Cb	Cr	Y	Cb	Cr
S01	9.3296	15.0736	11.5338	8.5185	14.8263	12.4571	9.3222	12.4174	12.9084
S02	9.3530	15.5290	14.2688	10.3549	16.7462	16.5045	8.0752	11.7164	11.7153
S03	10.0783	11.6349	13.4709	10.2610	12.4442	12.6144	9.7153	14.6438	16.6390
S04	9.9165	15.5669	14.8953	7.6580	15.0567	15.5930	8.2285	13.7008	14.8733
S05	9.2302	14.1618	14.5752	9.7136	12.7344	12.1320	10.2523	13.7524	13.8768
S06	8.0488	11.4877	15.5186	8.3074	15.0308	12.7250	8.3072	14.9717	14.8359
S07	8.2198	12.3925	15.4546	8.7685	15.1708	11.5467	9.7965	15.6217	14.2683
S08	10.1595	13.7344	12.6708	9.1436	11.4080	14.4573	8.0660	13.1013	14.8839
S09	7.5860	15.7875	14.4937	10.3282	12.5287	15.1002	8.3625	14.9721	14.2633
S10	8.9697	15.8244	11.9890	8.7532	12.3442	14.2796	7.7733	13.4970	15.3263
S11	8.0038	11.7881	11.1527	10.4492	15.0070	13.5544	9.2286	16.0516	14.1350
S12	10.4360	15.8530	14.7204	8.4044	16.0664	14.8667	9.5501	15.9975	16.9622
S13	9.6381	15.7858	13.5001	9.6033	13.0668	14.8857	9.1398	12.5386	12.3121
S14	9.0014	13.4269	13.3996	9.4990	15.6831	15.0741	8.7772	14.6808	11.6348
S15	8.9133	15.0014	15.5236	9.1174	15.0520	14.8147	9.4333	14.4935	11.6582
S16	7.6789	11.7094	14.0493	9.5943	11.0403	16.6710	9.4429	14.2444	11.3815
S17	9.5459	13.1088	14.0883	9.4996	14.6130	12.2536	9.5371	16.2196	13.4275
S18	7.6273	15.5787	15.2972	8.0344	13.3206	15.2557	9.4074	16.2196	13.6902
Average	**8.9853**	**14.0803**	**13.9223**	**9.2227**	**14.0078**	**14.1548**	**9.0231**	**14.1783**	**13.8218**

Another subjective perceptual metric is required to assess the quality of the proposed approach accurately. The structural similarity index (SSIM) [28] metric uses only luminance component for calculation, since the human visual system is more sensitive to luminance than chrominance components, and it evaluate the structural distortion between two frames. Table 7 illustrates PSNR (column 2 and 3) and SSIM (column 4) results of encrypted frames of PartyScene sequence decoded in low delay mode at equidistant QP values (column 1). The obtained results show clearly that for each QP value, a high visual degradation is obtained approved by the PSNR values that are less than their original values, which confirm the full distortion obtained using the proposed SE since all values are less than 14 dB. Additionally, all SSIM obtained values are less than 0.6, which signifies that there is no visual structural correlation can be established between original and encrypted frames.

Based on the above results, it is clear that the proposed selective encryption algorithm ensures a good confidentiality level with very acceptable performances according to the criterions detailed in [31].

4.1.2. Encryption space

Encrypted bitstream generated by the proposed SE have the same size as the plain bitstream because the protected bins were entropy-coded in bypass mode of BAC using 19 bits for coding each bins sequence (i.e., encrypted as well as plain sign/suffix are encoded with just 19 bits). As a result, the change after encryption will affect only theses 19 bits of encrypted sign/suffix, and the total size of

encrypted bitstream will be maintained without additional overhead.

Table 6. PSNR average of 100 frames per sequence encrypted in all tested modes at QP=18.

	Intra only mode			Low delay mode			Random access mode		
	Y	Cb	Cr	Y	Cb	Cr	Y	Cb	Cr
S01	9.3353	17.4925	16.8796	9.7456	18.2675	18.6889	9.5724	17.5338	22.0385
S02	9.5315	16.6851	19.2407	9.1159	18.9106	19.9781	10.6049	20.1064	18.6508
S03	11.2374	17.5660	19.2479	10.2636	17.3849	20.2468	10.2112	17.3031	18.4774
S04	9.1786	16.5475	19.9501	10.9036	16.6243	18.4712	9.3796	21.2654	23.2280
S05	9.6062	19.9787	18.6209	10.5777	18.1401	19.2777	10.9548	19.4114	23.7490
S06	9.1344	16.4243	16.3181	9.2247	17.6311	18.5954	9.2515	19.3014	21.4338
S07	10.1043	20.1852	20.1524	9.2022	19.5465	16.1164	9.7352	16.9493	18.1706
S08	9.0332	17.7956	19.6017	10.9431	19.5113	18.0094	9.5934	16.4969	16.7537
S09	11.2430	16.2133	17.2868	11.2628	19.0083	18.9084	10.3272	17.4039	23.1252
S10	9.4916	17.5407	18.4465	10.3344	16.6008	18.3454	9.2287	20.7221	20.9701
S11	9.2334	19.3118	20.4315	9.2729	16.0970	17.6754	10.0133	21.0109	24.0589
S12	9.7684	19.5761	19.2206	11.0645	18.5193	20.2171	9.2621	19.8480	16.5187
S13	10.1401	18.4521	19.7754	9.8452	17.3537	19.7329	9.2807	19.9885	20.9695
S14	9.2542	19.0880	17.9497	9.7349	20.2273	19.8209	10.9611	17.2644	18.4234
S15	11.4885	20.0213	18.1178	10.8658	20.4141	17.6764	9.7289	19.1683	23.0357
S16	9.8302	16.2466	18.5232	9.0258	17.2898	18.6693	10.5088	20.4585	17.6234
S17	9.7434	17.3665	17.2109	9.1211	19.6037	19.9265	11.4111	18.2211	19.7615
S18	9.1551	16.2079	19.3706	10.6698	20.0325	20.2008	10.0812	21.4364	19.3440
Average	**9.8060**	**17.9277**	**18.6858**	**10.0652**	**18.3979**	**18.9198**	**10.0059**	**19.1050**	**20.3518**

We define the encryption space ES as the percentage of bits chosen by our SE to be modified after encryption. In following experiments, we have measured the ES resulting from the encryption of 100 frames of each sequence in all tested modes.

Table 7. Variation of PSNR and SSIM of decode frames in low delay mode OF PARTYSCENE sequence over equidistant QP values.

QP	PSNR Y(dB)		SSIM	Encrypted bits per frame	ES(%)
	Original	Encrypted			
12	52.45	8.02	0.019	232011	21.12
18	48.75	8.75	0.12	198255	17.64
24	42.11	9.15	0.26	117525	18.15
30	37.70	10.96	0.20	80657	15.75
36	33.95	10.85	0.32	52336	13.96
42	27.15	12.63	0.28	21045	13.12

Table 8 gives us the estimations found for the encryption space (in %) for encrypted bitstream compressed at QP=18. It is clear that the encryption spaces for each sequence in both low delay and random access modes are close to each other and they change from sequence to another. We note that all results in these modes are less than 38%, which means that at least 38% of bits can provide a high visual degradation.

In Table 7, the impact of QP on ES is examined (column 6), where the latter will represent the average of encrypted bits per frames. It appears clearly from the experience that the ES varies inversely to both QP values and bitstream sizes. The lowest value of ES is approximately 13%. In second experience, we found that the average of encrypted bits per frames is 21045 at QP=42, which confirms the reliability of our SE over all compression's levels.

Table 8. Encryption space ES(%) obtained from the encryption of the benchmark sequences at QP=18

Sequence	Intra only	Low delay	Random access
S01	30.6849	17.8418	19.1556
S02	30.6581	20.0715	16.8726
S03	31.7781	22.6669	16.9228
S04	36.4829	20.9676	19.5627
S05	33.0044	18.2084	21.4779
S06	32.4337	21.2471	17.9543
S07	30.5235	23.1099	18.9205
S08	37.2359	19.6139	20.4343
S09	32.4634	19.5909	22.7974
S10	36.1781	17.6444	16.3734
S11	34.2303	15.6055	21.0829
S12	32.2012	16.5468	19.9108
S13	36.4560	15.7904	18.6830
S14	35.6712	18.9397	22.5161
S15	32.9521	15.0793	18.3409
S16	35.9188	22.7777	16.5213
S17	36.6368	20.4633	20.3833
S18	32.6485	15.0121	20.3040

4.2. Time processing

Experimental simulations are performed on Intel 2.3GHz Dual-Core T4500 processor with 3 GB of random access memory (RAM). To measure the time of processing of every process accurately since this last is often affected to noise during its running, we performed each process (encoding/decoding) several times, and we calculated later the average time of the process execution. The overhead time resulting from the proposed SE plays a crucial role in determining the rapidity of the proposed approach. Table 9 gives the processing time results in seconds of encoding/decoding process with and without encryption/decryption using the Kimono sequence at QP=18, and we evaluated the corresponding computational time at different number of frames (10,30,50,100) in low delay mode.

Table 9. Processing time overhead (in seconds) for Kimono sequence encrypted in low delay mode at QP=18.

	Encoding time				Decoding Time			
Number of frames	Without SE	With SE	Difference time	σ_e	Without SE	With SE	Difference time	σ_d
10	4930.25	4931.15	0.90	2.12×10^{-2}	58.15	58.35	0.20	0.12×10^{-4}
30	14850.36	14851.60	1.24	3.82×10^{-2}	168.23	168.91	0.68	0.62×10^{-4}
50	24550.91	24553.06	2.15	4.23×10^{-2}	280.37	282.22	1.85	1.32×10^{-4}
100	48609.12	48613.03	3.91	7.64×10^{-2}	518.24	521.86	3.62	3.96×10^{-4}

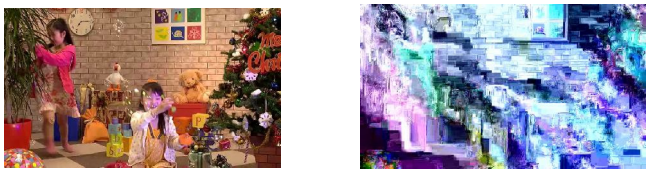
Both σ_e and σ_d represent the standard deviation measures of encoding and decoding process respectively which help us for estimating the accuracy of difference times. It is clear that the difference times resulting from encryption and decryption with the proposed approach are negligible compared to the overall possessing time. Hence, we conclude the suitability of proposed SE for real-time application.

4.3. Security analysis

4.3.1. Key space analysis

In order to test the robustness of the algorithm, we assumed that the attacker knows well the core of the SE algorithm, and the security of the proposed algorithm depends only on the secret key knowledge. The proposed selective encryption algorithm uses 128 bits for K_1, and 127 bits for K_2. The key space can then contains $2^{128+127} = 2^{255}$ possible keys. Therefore, we consider that the key space is sufficiently large to permit robustness against exhaustive key search.

4.3.2. Key sensitivity analysis



(a) (b)

Fig. 8. Decoding results of PartyScene encrypted bitstream encoded in Intra only mode at QP=20: (a) decrypted frame with original key, (b) decrypted frame with wrong key.

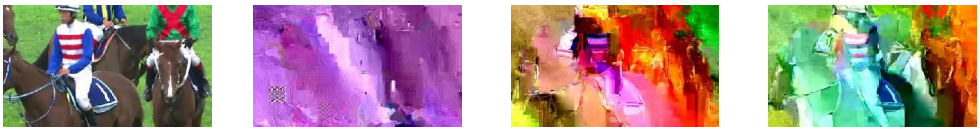
In this experiment, we suppose that the attacker is able to guess an approximate key as close as possible to the original key except in one or two bits, and we test if he can attack the proposed SE by decrypting bitstream with a wrong key. $K_1$is the main key of our approach since the second key is used only for protecting the latest few bits. Fig. 8 shows two decoded frames of PartyScene at QP=20 in Intra only mode, when the first frame is encrypted using original keys $K_1 = FFFFFFFFFFFFFFFF$, and the second is decrypted with $K_1 = FFFFFFFFFFFFFFFE$. The perceptual content of attacked frame with wrong K_1 is too low which signifies that the proposed approach is strong against such type of attack, and the proposed approach is highly sensitive to key variations.

4.3.3. Security against known-plaintext and correlation attacks

We suppose in following experiments that the attacker knows the core of the proposed SE, and attempts to broke it in order to improve the perceptual quality of attacked bitstream.

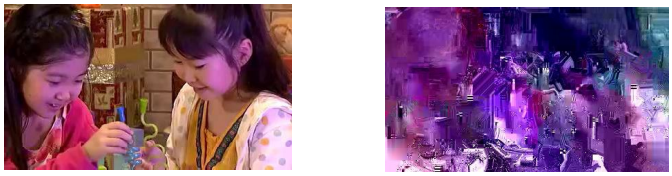
In the first experiment, we evaluate the robustness of the proposed SE against known *plaintext attacks* by guessing the encrypted bins of *QTCs*' ciphertext on the basis of their known values in corresponding plaintext. The bits of both signs and levels are either 0 or 1 and known to everyone. The attacker can retrieve them by a brute force attack, and attempt later to replace the encrypted bits with other ones chosen at random in order to make the perceptual content watchable.

The protection of *coeff_abs_level_remaining* suffixes makes the proposed SE strong against this type of attack. To illustrate this fact, we applied the experiments on the first frame of RaceHorses sequence encoded in low delay mode by replacing the encrypted bins of sign as well as TR_p /EG_{p+1} suffixes by a constant value of zero and we measured the corresponding PSNR values. In different level of visual



(a) (b) (c) (d)

Fig. 9. Sensitivity of the proposed SE against known plaintext attacks : (a) original frame of RaceHorses sequence encoded in low delay mod,(b) attacked result at QP=18 with PSNR(Y)=7.15,(c) attacked result at QP=28 with PSNR(Y)=15.28,and (d) attacked result at QP=34 with PSNR(Y)=16.05



(a) (b)

Fig. 10. Robustness of the proposed approach against *Correlation-based attacks*: (a) original sequence encoded in low delay mode, (b) attacked result with PSNR(Y)=8.16.

quality (low, medium, and high), a perceptual distortion is obtained as is illustrated in Fig. 9 with low PSNR values which means the strength of the proposed approach.

Since CABAC syntax elements are entropy coded separately, and sign and *QTCs* suffixes are protected independently in the proposed SE, the proposed become strong against *Correlation-based attacks*. Indeed, we suppose in a second experiment that the attacker can reach only to the plaintext of signs, but cannot obtain the entire plaintext of the QTCs' suffixes. The attacker cannot recover the original key since AES cipher is still robust against known-cipher attack. Nevertheless, he attempts by this experiment to improve the visual content of attacked bitstream. Fig. 10 presents both decoded frames of original and attacked bitstream encoded in low delay mode at QP=24. It is obvious that the visual content is still ambiguous with high visual degradation.

4.4. Comparative study and analysis

The global scheme of SE during entropy coding consists firstly of searching the syntax elements that do not alter the format compliancy and compression efficiency of encrypted bitstream. Later, the selected data will be protected by strong cipher. In the case of HEVC standard, SE can be performed by protecting syntax elements encoded in bypass mode.

In order to manifest the superiority of our approach, we have compared our approach with recent works on SE-HEVC over different criteria. All compared SE approaches employ AES cipher except [22]. Hofbauer *et al.* proposed in [22] to encrypt directly the bits of signs of *ACs* coefficients from bitstream. However, HEVC transmits the signs of each 4×4 sub-block together to BAC engine as is described in Fig. 3, which means there are no specific bits for *ACs* signs. Accordingly, this work needs

to be fixed according to [1]. Nevertheless, we can achieve this purpose by encrypting the bins of *ACs* signs of each transform block (or for each 4×4 sub-block) before transmitting them to BAC engine.

Table 10. Comparison analysis between selective encryption algorithms.

ENCRYPTION ALGORITHM	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
Shahid *et al.* [12]	Yes	QTCs levels + QTCs signs	No	AES+CFB	Yes	No	high	H.264/AVC
Asghar *et al.* [20]	Yes	CAVLC and CABAC syntax elements	No	AES+CFB	Yes	No	high	H.264/SVC
Wallendael *et al.* [21]	Yes	syntax elements (signs, MVDs, …)	No	AES+XOR	Yes	Yes	medium	HEVC
Hofbauer *et al.* [22]	Yes	Bits of QTCs signs	No	Not mentioned	Yes	No	low	HEVC
Ouamri *et al.*[23]	Yes	TR_p codes	No	AES	Yes	Yes	high	HEVC
Memos *et al.*[24]	No	QTCs levels	No	AES	Yes	Yes	high	HEVC
Shahid *et al.* [10]	Yes	QTCs levels + QTCs signs	Yes	AES+CFB	Yes	No	high	HEVC
Shahid *et al.* [11]	Yes	QTCs levels + QTCs signs	No	AES+CFB	Yes	No	high	HEVC
Proposed algorithm	**Yes**	**coeff_abs_level_remaining suffix + QTCs signs**	**No**	**AES-CBC**	**Yes**	**Yes**	**high**	**HEVC**

C_1: compression ratio maintained, C_2:encryption domain, C_3 :Context modeling,C_4 :Encryption algorithm, C_5:Compression dependency.C_6: format compliancy of encrypted bitstream according to the standard draft of HEVC. C_7: visual degradation of encrypted bitstream. C_8: format of video coding standard.

Regarding the visual quality of encrypted bitstream, the approach of [22] has the minimal impact on perceptual quality of encrypted bitstream, which is the main purpose of transparent encryption. Wallendael *et al.* studied in [22] the impact of encryption of syntax elements on visual quality and compression efficiency of encrypted bitstream. They found that by encryption of both MVD signs and residual signs, the visual quality could be distorted by delta PSNR of 22.1 dB [22, Table IV]. The averages of PSNR values obtained by Shahid *et al.* in [10]-[11] are 9.67 dB for the luminance component, 15.82 dB and 17.23 dB for U and V components respectively. In the case of our approach, PSNR values are 9.22 dB for Luminance component, 14.15 dB and 14.17 dB for U and V respectively [10, Table IV], [11, Table IV]. We conclude that our proposed approach provides sufficient perceptual security in comparison to compared works.

Table 11. Encryption quality comparison between our approach and [10]-[11] for *kimono* sequence encryption at qp=18 for different bit-shifts.

Left bit shift	0	2	4	6
[10]	9019	8943	8472	8540
[11]	7002	7100	6879	7199
Our approach	11356	10825	9265	9012

The encryption quality (EQ) represents the total changes in pixels values between the original frames and the encrypted ones, and it can be calculated by [33, eq. (1)]. In Table 11, a comparative analysis in terms of encryption quality is done between the two works [10]-[11] and ours SE. we have found that EQ values of our approach are all superior to those of [10] and [11].

In addition, our approach can be performed faster than [10] since the differences in time processing of our approach are approximately 0.03 and 0.04 for encryption and decryption respectively, while in [10],

these computational times are 31.68 and 0.04 seconds respectively.

We have also compared our approach with recent works on SE-H.264 such as [12] and [20]. Transform coding in H.264 is done either by CAVLC or by CABAC. Compared to HEVC, non-zero QTCs in H.264 are binarized by unary/EG_0 codes before coding by BAC. The profound difference existing between CABAC structure of H.264 and those of HEVC means that each video standard needs its own suitable SE.

5. Conclusion

In summary, we have seen that the previous works on SE-HEVC based on QTCs encryption become not suitable since both codes class and threshold of transition codes are totally changed in the standard draft of HEVC. So, it is needed to find a compliant encryption space conform to the latest novelties reached in the newly approved standard draft of HEVC.

We have presented in this paper one of the first works based on QTCs encryption allowing protection of HEVC bitstream in format compliant manner. The main specific contribution in this paper is what exactly can be encrypted from levels of QTCs. We have proposed a strategy for QTCs selection that preserve the compression efficiency of HEVC bitstream according to the latest modification reached in the approved HEVC standard draft. Both visual and security tests confirm the efficiency of the proposed approach in terms of visual content degradation, time processing and robustness against replacement attacks.

In our view, the present findings of this study constitute an excellent initial step towards HEVC application security chiefly network applications (video conference, streaming…) since the bitrate as well as format compliancy of encrypted bitstream is conserved by our approach.

References

[1]. High efficiency video coding, ITU-TRec.H.265 and ISO/IEC 23008-2 (MPEG-H, Part 2), Apr.2013, version 1.

[2]. G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[3]. T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[4]. D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 620–636, Jul. 2003.

[5]. T. Nguyen, D. Marpe, H. Schwarz, and T. Wiegand, "Reduced-complexity entropy coding of transform coefficient levels using truncated golomb-rice codes in video compression," in *2011 18th IEEE International Conference on Image Processing (ICIP)*, 2011, pp. 753–756.

[6]. S. Golomb, "Run-length encodings (Corresp.)," *IEEE Transactions on Information Theory*, vol. 12, no. 3, pp. 399–401, Jul. 1966.

[7]. B. Bross, W.-J. Han, J.-R. Ohm, G. J. Sullivan, and T. Wiegand, "High Efficiency Video Coding (HEVC) Text Specification Draft 6," Tech. Rep., ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), San Jose, CA, USA, 2012, Doc. JCTVC-H1003.
[8]. J. Sole, R. Joshi, N. Nguyen, T. Ji, M. Karczewicz, G. Clare, F. Henry, and A. Duenas, "Transform Coefficient Coding in HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1765–1777, Dec. 2012.
[9]. C. Kim, J. Kim, and J. Park, "Simplification of Golomb-Rice Parameter Update," Tech. Rep., ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Geneva, Switzerland, 2012, JCTVC-10124.
[10]. Z. Shahid and W. Puech, "Visual Protection of HEVC Video by Selective Encryption of CABAC Binstrings," *IEEE Transactions on Multimedia*, vol. 16, no. 1, pp. 24–36, Jan. 2014.
[11]. Z. Shahid and W. Puech, "Investigating the structure preserving encryption of high efficiency video coding (HEVC)," 2013, p. 86560N.
[12]. Z. Shahid, M. Chaumont, and W. Puech, "Fast Protection of H.264/AVC by Selective Encryption of CAVLC and CABAC for I and P Frames," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 5, pp. 565–576, May. 2011.
[13]. T. Nguyen, P. Helle, M. Winken, B. Bross, D. Marpe, H. Schwarz, and T. Wiegand, "Transform Coding Techniques in HEVC," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 978–989, December 2013.
[14]. T. Nguyen, P. Helle, M. Winken, B. Bross, D. Marpe, H. Schwarz, and T. Wiegand, "Corrections to Transform Coding Techniques in HEVC" *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 6, pp. 1194–1195, Dec. 2014.
[15]. Y. H. Tan, C. Yeo, H. L. Tan, and Z. Li, "On residual quad-tree coding in HEVC," in *2011 IEEE 13th International Workshop on Multimedia Signal Processing (MMSP)*, 2011, pp. 1–4.
[16]. V. Sze and M. Budagavi, "High Throughput CABAC Entropy Coding in HEVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1778–1791, Dec. 2012.
[17]. J. Wang, Y. Fan, T. Ikenaga, and S. Goto, "A partial scramble scheme for H.264 video," in *7th International Conference on ASIC, 2007. ASICON '07*, 2007, pp. 802–805.
[18]. H. Sohn, E. T. AnzaKu, W. De Neve, Y. M. Ro, and K. N. Plataniotis, "Privacy Protection in Video Surveillance Systems Using Scalable Video Coding," in *Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, 2009. AVSS '09*, 2009, pp. 424–429.
[19]. Y. Kim, S. H. Jin, T. M. Bae, and Y. M. Ro, "A selective video encryption for the region of interest in scalable video coding," in *TENCON 2007 - 2007 IEEE Region 10 Conference*, 2007, pp. 1–4.
[20]. M. N. Asghar, M. Ghanbari, M. Fleury, and M. J. Reed, "Sufficient encryption based on entropy coding syntax elements of H.264/SVC," *Multimed Tools Appl*, pp. 1–27, Jul. 2014.
[21]. G. Van Wallendael, A. Boho, J. De Cock, A. Munteanu, and R. Van de Walle, "Encryption for high efficiency video coding with video adaptation capabilities," *IEEE Transactions on Consumer Electronics*, vol. 59, no. 3, pp. 634–642, Aug. 2013.
[22]. H. Hofbauer, A. Uhl, and A. Unterweger, "Transparent encryption for HEVC using bit-stream-based selective coefficient sign encryption," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 1986–1990.
[23]. M. Ouamri and K.M. FARAOUN, "Robust and fast selective encryption for HEVC videos", JOURNAL OF COMMUNICATIONS SOFTWARE AND SYSTEMS, VOL. 10, NO. 4,pp. 221-229, DECEMBER 2014.
[24]. V. A. Memos and K. E. Psannis, "Encryption algorithm for efficient transmission of HEVC media," *J Real-Time Image Proc*, pp. 1–10, May 2015.
[25]. J. Daemen and V. Rijmen, *The Design of Rijndael: AES - The Advanced Encryption Standard*, Softcover reprint of the original 1st ed. 2002 edition. Berlin, Heidelberg: Springer, 2013.
[26]. M. J. Dworkin, "SP 800-38A Addendum. Recommendation for Block Cipher Modes of Operation: Three Variants of Ciphertext Stealing for CBC Mode," National Institute of Standards & Technology, Gaithersburg, MD, United States, 2010.
[27]. HM reference software: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/
[28]. F. Bossen, "Common HM test conditions and software reference configurations," document JCTVC-L1100 of JCT-VC, Geneva, CH, Jan. 2013
[29]. G. Bjøntegaard, "Calculation of average PSNR differences between RD curves," in ITU-T SG 16 Q. 6 Video coding Experts Group (VCEG), Document VCEG-M33, Austin, TX, Apr. 2-4, 2001.

[30]. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[31]. L. Dubois, W. Puech, and J. Blanc-Talon, "Reduced selective encryption of intra and inter frames of H.264/AVC using psychovisual metrics," in *2012 19th IEEE International Conference on Image Processing (ICIP)*, 2012, pp. 2641–2644.
[32]. TU-T and ISO/IEC JTC 1, Advanced video coding for generic audiovisual services, ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), 2011.
[33]. H. Ahmed, H. Kalash, and O. Allah, "Encryption efficiency analysis and security evaluation of RC6 block cipher for digital images," in *Proc. Int. Conf. Electrical Engineering*, Apr. 2007, pp. 1–7.

Mokhtar Ouamri / orcid.org/0000-0002-3072-8647

Received his engineer degree from the University of Sciences and Technologies of Oran (USTO), Oran, Algeria, in 2007, and the M.S. degree from the University of Sciences and Technologies of Oran (USTO), Oran, Algeria, in 2010, both in computer science. Since December 2011, he is PhD candidate in computer science, at Djillali Liabes University (UDL), Algeria. He is currently Assistant Professor at University of Ibn-Khaldun, Tiaret, Algeria. His research interests are in the fields of multimedia compression/security, image/video processing and analysis, video surveillance (detection, tracking, event detection, and storage), and multimedia communication.

Kamel Mohamed Faraoun

Received his M.S. degree in computer science from Djillali Liabes University (UDL) of Sidi-Bel-abbes, Algeria in 2002, and his Ph.D degree in computer science, in 2006, and his Habilitation à Diriger des Recherches (HDR) degree, in 2009, at the same university. His current research areas include computer security systems, cryptography, multimedia communications, genetic algorithms, cellular automata, evolutionary programming and information theory. He is currently Associate Professor at computer science department of UDL University. Dr. Faraoun is a member of the Evolutionary Engineering and Distributed Information Systems Laboratory (EEDIS).

Résumé :

L'information multimédia occupe une place incontournable dans notre vie quotidienne avec un succès florissant et incontestable dans les marchés actuels de la technologie numérique. La vidéo est l'une des informations multimédias les plus utilisées. On la trouve dans des applications très variés en ou hors ligne comme la vidéo conférence, VOD (vidéo à la demande), et dans les réseaux sociaux comme celles de streaming. Une communication sûre déployant la vidéo numérique nécessite le passage par des étapes pionnières comme la compression (ou le codage de source) pour réduire la quantité transmise et d'extraire l'information pertinente à transporter, le codage pour combattre les erreurs de transmission et la protection de l'information par des mesures de sécurité logicielle et/ou matérielle comme la signature numérique, la stéganographie, le tatouage numérique, et la cryptographie par des approche de chiffrement.

HEVC (High Efficiency Video Coding) est la dernière norme de codage vidéo. Elle apporte une architecture hybride de codage permettant ainsi la compression de vidéos à haute définition. En effet, cette norme est attendue à être le futur codec de l'ère Ultra HD. Comme la vidéo HEVC représente le résultat d'une compression de grande quantité de données visuelles, le maintien de la taille de flux binaire compressé s'avère inévitable lors de la conception d'un système de crypto-compression pour la protection de vidéos HEVC. Et ceci pourra être atteint par l'inclusion de module de chiffrement durant CABAC (Context-adaptive binary arithmetic coding) qui est le seul codeur entropique utilisé depuis HEVC.

Dans cette thèse, nous avons proposé deux approches de chiffrement sélectif pour les vidéos HEVC, où nous avons chiffré des éléments syntaxiques relatives aux coefficients fréquentiels quantifiés. Dans notre première approche, les codes de Golomb-Rice récemment introduits dans HEVC et les signes de coefficients non-nuls sont protégés. Alors dans la deuxième approche, nous avons choisi tous les codes utilisés pour le codage de coefficients non-nuls avec ses signes pour le chiffrement dans un contexte bien étudié. Les deux approches proposées génèrent des flux binaires HEVC cryptés conformes à la norme de codage avec une taille identique aux flux binaires clairs. En conséquence, les contributions présentées dans cette thèse constituent en effet un premier pas vers la sécurité de la norme HEVC.

Mots clés : *HEVC, chiffrement sélectif, CABAC, codage vidéo.*