画像・映像フォーマット準拠暗号化と
その再符号化可能性・調整可能性・汎用性に関する研究

2022年　3月

清水 恒輔

画像・映像フォーマット準拠暗号化と
その再符号化可能性・調整可能性・汎用性に関する研究

清水 恒輔

システム情報工学研究科

筑波大学

2022年　3月

目次

図目次

表目次

第 1 章

はじめに

1.1 本論文の背景・目的

最古の暗号技術は，紀元前 3000 年に古代エジプトで用いられたヒエログリフであると言われている [1]．古代ではその他に，スキュタレー暗号やシーザー暗号，上杉暗号などが人の情報伝達を秘密裏に実現する手段として用いられてきた．さらに，1900 年代の第一次世界大戦時には，ドイツが軍事目的で ADFGVX 暗号やエニグマなどの暗号を開発した．これらは物理領域での情報伝達手段として用いられたが，ディジタル化が進みコンピュータによる暗号の解読が行えるようになると，もはや ‘暗号化器を隠すことによるセキュリティ’ は安全視されなくなった．1949 年，ディジタル通信における解読不可能な暗号として，情報理論の父 C.E. シャノンはワンタイムパッド [2] を提唱したが，平文と同長の鍵（キー）が必要になることで，実用的な場面は限られる．キーの長さを固定長かつコンパクトにしつつ，1970 年代当時のセキュリティを実現する数理暗号として開発された DES: data encryption standard は，56 ビットというキーの長さが総当たり攻撃に対する脆弱性の懸念を招くため，現在では利用推奨されていない．DES の拡張である Triple DES ではキー長の問題を解決したものの，DES を超える暗号として後に開発された AES: advanced encryption standard [3] が現在最も信頼されている世界標準の暗号技術の一つである．しかし，DES や AES は暗号化・暗号復号（平文化）に同一のキーを用いる対称鍵暗号系に分類され，見ず知らずの他人にキーを共有する必要があり，鍵配送問題が浮上する．鍵配送問題を解決しつつセキュリティを実現する手段として，RSA: Rivest Shamir Adleman 暗号 [4] や DH: Diffe-Hellman 鍵共有などの非対称鍵暗号系の技術が盛んに研究された．このような高信頼な暗号技術の確立によって，現代のコンピュータやディジタルカメラ，スマートスピーカなど ‘モノ’ が人とインターネットで接続される IoT: internet of things の土壌は支えられている．

暗号技術と別に圧縮符号化もまた不可欠な技術である．インターネット上で（ディジタル信号として）通信されるテキストや画像，映像コンテンツを利用することで，人の情報伝達のインフラとして IoT サービスが機能していることは言うまでもない．また，やりとりされる画像・映像コンテンツの高

解像度化で，より人の知覚を正確に再現できる ‘解像度革命’ が広まっている．しかし，非圧縮な高解像度コンテンツでは大量保存・高速通信が困難となるため，一つ一つのファイルサイズを劇的に削減しつつ，視覚品質を極力劣化させないような圧縮符号化技術が開発され続けている．画像符号化標準規格としては JPEG シリーズがあり，世界で初めて規格化された JPEG はデファクトスタンダードとして 20 年以上前から世界中に普及している．その後継として JPEG 2000 [5] や JPEG XR [6] が開発され，JPEG 2000 を軽量化・高速化した JPEG XS [7]，また高ダイナミックレンジ画像符号化のための JPEG XT [8]，点群やホログラフィックデータ符号化のための JPEG Pleno [9] など，その種類は多岐にわたる．また，映像符号化標準規格としては MPEG・H.26x シリーズがあり，地上ディジタルテレビ放送やインターネット映像配信に MPEG-2 [10]，H.264/AVC [11]（以下，AVC），H.265/HEVC [12]（以下，HEVC）などが既に採用され，さらには H.266/VVC [13]（以下，VVC）が 2020 年 7 月に第 1 版の標準化が完了している．

画像・映像コンテンツがスマートに通信される中で，とりわけ Twitter [14]・Facebook [15] を代表とする SNS: social networking service や Hulu [16]・Netflix [17] などを代表とする SVOD: subscription video on demand では様々な度合のセキュリティが要求されている．SNS では，不特定多数の視聴者がユーザのアップロードしたコンテンツを視聴できるため，非公開設定ではなく適度な暗号化（秘匿化）による保護を要求される場面が多々存在する．また，SVOD では，多くのユーザに映像コンテンツへの興味を持たせて購買意欲を促すために，無料コンテンツの内容を適度に秘匿化することが要求される．セキュリティのみを重視すれば，AES・RSA 暗号などの強力な暗号を用いて対象のコンテンツフォーマットを元と異なるものに変換すれば良い．しかしながら，SNS や SVOD などでは，強力な暗号化によって異なるフォーマットに変換された画像・映像コンテンツがそもそもアップロード非対応であることや，アップロードサポートされていたとしてもまずその暗号化ファイルを平文化してから試聴させる必要があることなど，コンテンツの送信にいくつかの課題が残されている．換言すれば，前述したように適度に秘匿化した状態でのコンテンツをアップロードしたり・視聴したい者にとっては，強力な暗号化は不都合といえる．このような状況を解決するには，コンテンツフォーマットに準拠しつつ，画像・映像コンテンツを適度に秘匿化するフォーマット準拠暗号化が求められる．

フォーマット準拠暗号化には，現状で以下に記す三つの社会的要求が存在する：

要求 1 アップロードコンテンツのファイルサイズ増大に対するストレージ・通信効率化のため，コンテンツを再びファイルサイズ削減（再符号化）した後でもコンテンツの平文化品質を損なわない ‘再符号化可能性’.

要求 2 ある符号化規格に特化したフォーマット準拠暗号化は他の規格に準拠せず，他の規格用に再設計する際の実装コスト・時間面で難が生じるため，あらゆるフォーマットに共通に適用できる ‘汎用性’.

要求 3 プロバイダによって異なるコンテンツに求める知覚劣化度合への対応のため，単一の暗号化の達成する知覚劣化度合を調整できる ‘調整可能性’.

SNS では非可逆な再符号化を行っている．圧縮符号化後の暗号化 [18–21] には数理暗号に基づく手法のみならずフォーマットを維持できるフォーマット準拠暗号化がある．しかし SNS では元のビットストリームが再符号化によって非可逆に変換されるため，圧縮符号化後の暗号化ではこの再符号化への対応は困難である．圧縮符号化中の暗号化 [22–24] はフォーマットを維持したまま低計算量かつ適度な知覚劣化度合を実現できるものの，圧縮符号化後の暗号化と同じように途中の信号が再符号化によって非可逆に変換されるため，やはり再符号化への対応は難しい．一方で，圧縮符号化前の暗号化 [25–27] は再符号化の暗号化・平文化への影響を抑えるように設計可能であるため，SNS での再符号化にも対応している．

また，SVOD ではプロバイダごとにコンテンツの知覚劣化度合が異なる．異なる知覚劣化をそれぞれ満足するような暗号化手法はこれまでも多く提案され，軽微な保護 [28] から厳重な保護 [29] まで分類されてきたが，所望の知覚劣化度合ごとに暗号化手法を設計し直さなくてはならないのはコスト・実装時間などの面で難であり，単一の暗号化手法のみで様々な知覚劣化度合を実現できることが望ましい．こうした要求のために，近年では AVC・HEVC 圧縮映像保護のための調整可能暗号化 [30–32] が提案されている．これらの手法では，元映像シーケンスをどれほど暗号化するかという確率パラメータを用いて，圧縮映像ファイルに含まれる特有の構文要素（イントラ予測モード，動き補償差分成分など）を暗号化する．ただし，その暗号化対象の構文要素は圧縮画像ファイルに含まれておらず，画像符号化に用いることができない．とくに JPEG は画像符号化のデファクトスタンダードでありながら，その圧縮フォーマット JFIF: JPEG file interchange format に上記のような構文要素が含まれていない，すなわち上記従来法を用いれない．しかしながら，圧縮符号化中・後の暗号化は構造・模様情報ごとにかつそれらを同時に暗号化できるため，汎用性や調整可能性の実現ポテンシャルを秘めていると考えられる．

上記を受けて，本研究では大別して二つの手法を提案する：

提案 1 再符号化可能性に関して，コンテンツを符号化前で適用可能な，つまり既存の画像・映像符号化とは独立している暗号化.

提案 2 汎用性と調整可能性に関して，コンテンツの符号化中・後で適用可能かつコンテンツを適切な粒度で一部のみ暗号化したとしてもその安全性が担保されるような暗号化.

提案 1 のために，主に画像符号化のデファクトスタンダードとして根強く採用されている JPEG を各フレームに適用した映像符号化 Motion JPEG（MJPEG） [33] に着目し，MJPEG の符号化前段階で適用可能なピクセルキューブベース暗号化を設計する．そして，提案 2 のために，JPEG 圧縮画像の知覚劣化度合を柔軟に調整可能であり，JPEG のほかに HEVC などの映像符号化にも共通に適用可能な

ビットキューブベース暗号化を設計する．

1.2 本論文の構成

本論文の構成は以下となる．まず，フォーマット準拠暗号化の提案に必要な関連技術を 2 章で説明する．暗号化の適用部分となる画像・映像符号化の処理を JPEG を中心に説明し，暗号化の基礎技術や，暗号化に用いられる乱数，フォーマット準拠暗号化に対する攻撃，従来のフォーマット準拠暗号化，二進化について説明する．

3 章ではピクセルキューボイドベース暗号化について言及する．JPEG 委員会が JPEG シリーズで圧縮（符号化）されるコンテンツのセキュリティに着目した ‘JPEG Privacy & Security’ [34] という活動を進めていることからも，SNS ／クライドサービス上で通信される画像・映像のセキュリティが重要視されている．既存のフォーマット準拠暗号化のうち MJPEG に注目した手法にはピクセルブロックベース暗号化 [27] があり，この手法は MJPEG で符号化される映像フレームを $16i \times 16j$ $(i, j \in \mathbb{N})$ ブロックベースで暗号化する．暗号化後のブロックが MJPEG の符号化ブロックを跨がないために，ピクセルブロックベース暗号化では MJPEG 符号化効率を維持できるものの，暗号化フレームの各ブロック内はほぼ暗号化されず，元フレームブロック内の模様情報が残っているために，ジグソーパズル解読攻撃への脆弱性を撤廃できない．ブロック内の秘匿性を向上させて単一フレーム内での不正な復元を防止するために，3 章ではピクセルキューボイドベース暗号化を提案する．

4 章ではビットキューボイドベース暗号化について言及する．たくさんの JPEG フォーマット準拠暗号化 [22–24, 27, 35–44] が提案されているものの，これらは本研究での調整可能性への言及をないがしろにしている．これらの従来法では，暗号化が JPEG 符号化の歪みをほとんど／全く劣化させないということと，JPEG 圧縮画像ファイルのビットレートを維持する／ほとんど増加させないということに固執している．よって，4 章では JPEG 圧縮画像の知覚劣化度合を柔軟に調整可能な（調整可能性を持つ）ビットキューボイドベース暗号化を提案する．

最後に，5 章で本論文をまとめる．

第 2 章

関連技術

本章では，フォーマット準拠暗号化の提案に必要な関連技術を紹介する．

2.1 画像・映像符号化

2.1.1 JPEG

JPEG の符号化アーキテクチャは大まかに図 2.1 に示される流れとなっており，それぞれ順を追って説明する．

a) 色変換・色差間引

カメラのレンズを通してカラーフィルタアレイでキャプチャされ，デモザイク処理された RGB 成分のピクセルデータは YCbCr 成分に変換され，輝度（Y）・色差（Cb・Cr）の情報に非相関化される．ある画素の RGB ピクセル値 $[R\ G\ B]^\top$ は，変換後の YCbCr ピクセル値 $[Y\ Cb\ Cr]^\top$ に

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.168736 & 0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.081312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} \tag{2.1}$$

と変換される．ここで，$[0\ 128\ 128]^\top$ の項はオフセットであり，$\top$ は転置を表す．YCbCr 成分のうち，色差（CbCr）成分の画素は縦横に間引かれる（図 2.2）．この間引のうち，JPEG では $[4:2:0]$ オプションがデフォルトで使用され，$[4:2:0]$ オプションでは色差成分の画素を縦横に 2 間引きする．

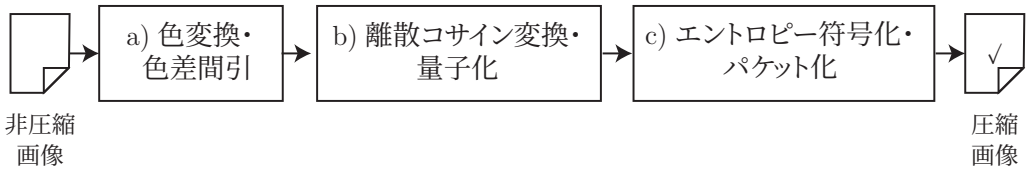


図 2.1　JPEG 符号化のフロー．

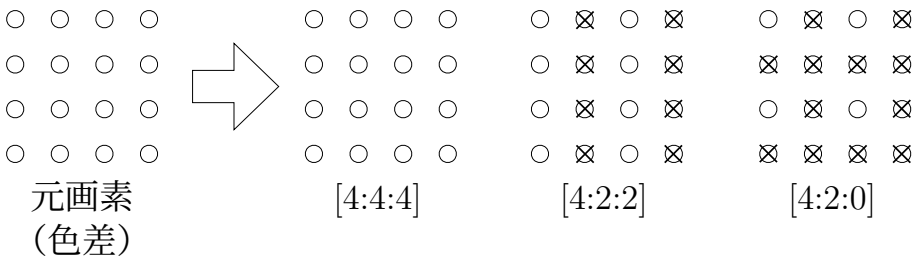


図 2.2　JPEG における色差間引.

色差間引の意義は，‘輝度の変化には敏感だが色の変化には鈍感であり，色差成分を時間方向で間引いたとしてもその変化がほぼ知覚されない’ という人間の視覚特性により色差成分を効率的に削減することにある.

b) 離散コサイン変換・量子化

間引後の YCbCr 成分の 8×8 ピクセルブロックごとに離散コサイン変換（DCT）で周波数変換し，その出力である 8×8 DCT 係数ブロックごとに，JPEG の Annex K カテゴリで規定された量子化テーブルと品質ファクタ Q で DCT 係数を量子化する．画像・映像フレームに対して最適な DCT はタイプⅡの DCT（DCT-Ⅱ） [45] であり，N 点のピクセル値 $\{p_i\}_{i=0}^{N-1}$ から N 点の DCT 係数 $\{c_j\}_{j=0}^{N-1}$ を得る DCT-Ⅱは

$$c_j = \sqrt{\frac{2}{N}} k_j \sum_{i=0}^{N-1} p_i \cos \frac{j(i+\frac{1}{2})\pi}{N},\ j = 0, 1, \cdots, N-1 \tag{2.2}$$

$$k_j = \begin{cases} 1 & j > 0 \\ 1/\sqrt{2} & j = 0 \end{cases} \tag{2.3}$$

で定義される．JPEG では $N = 8$ であり，画像の 8×8 画素ブロックを垂直方向に変換した後，水平方向に変換する（図 2.3）．8×8 DCT 係数ブロックの左上 1 成分は DC 係数，残り 63 成分は AC 係数と呼ばれる．ある 8×8 DCT 係数ブロック内の (i, j) 座標にある係数 $c_{i,j}$ は，JPEG の量子化テー

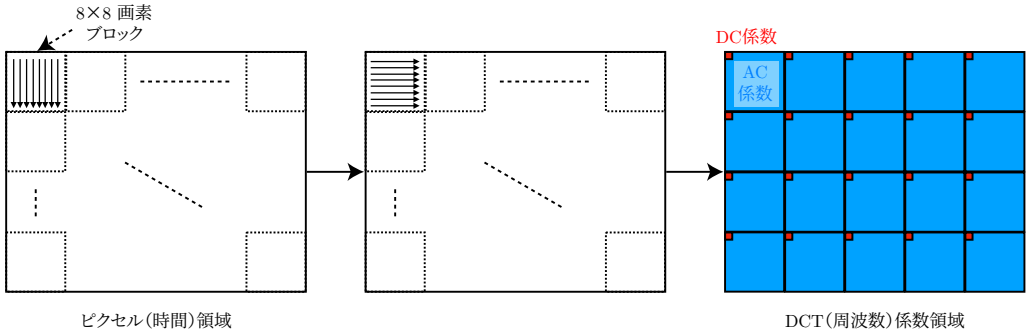


図 2.3　JPEG における離散コサイン変換の適用.

ブルによって

$$q_{ij} = \lfloor c_{ij}/d_{ij} \rfloor \tag{2.4}$$

$$d_{ij} = \begin{cases} \left\lfloor \frac{t_{ij} \times (5000/Q) + 50}{100} \right\rfloor & \text{if } Q < 50 \\ \left\lfloor \frac{t_{ij} \times (200 - 2Q) + 50}{100} \right\rfloor & \text{otherwise} \end{cases} \tag{2.5}$$

$$[t_{ij}]_{8\times 8} = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \tag{2.6}$$

と量子化される．ここで，(2.6) は Annex K で規定される JPEG の輝度成分量子化テーブル [33] であり，Q は品質ファクタである．本論文では，量子化後の DCT 係数 q_{ij} を QDCT: quantized DCT 係数と呼ぶ.

DCT・量子化を行う意義は，‘模様の滑らかな変化には敏感だが俊敏な変化には鈍感であり，俊敏な変化を表す周波数成分を量子化で削減してもその影響はほぼ知覚されない’ という人間の視覚特性により高周波の AC 係数を効率的に削減することにある．8×8 ブロック内で DC 係数（左上）に近い AC 係数ほど模様の滑らかな変化を，右下に近い AC 係数ほど俊敏な変化を表しているため，右下の AC 係数を効率的に削減するために，Annex K 量子化テーブルでは除算係数がジグザグスキャン順（図 2.4）に概ね昇順になっている．結果として得られる 8×8 QDCT 係数ブロックはジグザグスキャン順に降順の絶対値となり，人の目に鈍感な AC 係数ほど削減される（0 になる）．これまでの変換符号化では，画像の情報を非相関化・削減するものの，削減された各信号は依然として固定長のビット列として保持されている．また，$Q < 96$ の時の QDCT 係数は符号付の 10 ビットとして持たれており，当初の圧縮対象である 8 ビットのピクセルデータよりもビット方向に膨れ上がっているが，これらの固定長ビット列のゼロを削減することで，最終的な符号語データとして圧縮を行う.

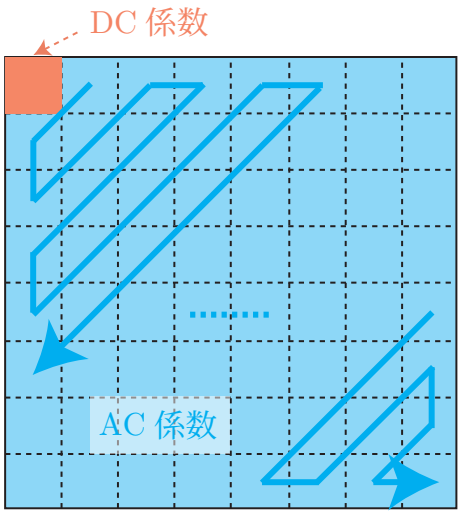


図 2.4　8 × 8 QDCT 係数ブロック内のジグザグスキャン.

c) エントロピー符号化・パケット化

QDCT 係数のうち，DC 係数と AC 係数は個別の手順でエントロピー符号化される．DC 係数は QDCT 係数ブロック間で互いに似通っているため，ブロック間で差分パルス符号変調（DPCM: differential pulse-code modulation）を適用することにより，その値（レベル）をさらに削減する．ここで，削減後の DC 係数値は DC 差分値と呼ばれる．AC 係数は量子化によって大量のゼロ値になっているため，ランレングス符号化（RLE: run-length encoding）を用いてある非ゼロ AC 係数までのゼロランレングスとその非ゼロ AC 係数値のペア（ランレベルペア）としてジグザグスキャン順（図 2.4）にまとめあげる．そして，DC 差分値と AC 係数ランレベルペアのカテゴリをハフマン符号化し，そのハフマン符号語と QDCT 係数の可変長ビット列が結合された可変長符号語（以下，結合符号語）を JFIF ファイル構造内の ECS: entropy-coded segment 部分に格納する．

このような可変長のビット列は，固定長ビット列をビット長付きの可変長ビット列に変換後，そのビット長と可変長ビット列をバイト単位で ECS セグメント内に格納（パケット化）される．DC 差分値のビット長（カテゴリ）は直接ハフマン符号化され，そのハフマン符号語と DC 差分値ビット列の結合符号語が JFIF ファイルの ECS セグメント内に格納される．カテゴリと DC 差分値ビット列からなる結合符号語は依然として可変長ビット列の形式となっているが，ある結合符号語の前後にある他の結合符号語と連結・分離させることで，これらの結合符号語をバイト単位で ECS セグメント内に格納される．また，AC 係数のランレベルペアは当該の (ランレングス, 非ゼロ AC 係数ビット長) の情報を AC 係数のカテゴリとしてハフマン符号化し，直後に非ゼロ AC 係数のビット列を連結した結合符号語として ECS セグメント内に格納される．たとえば 5 つのゼロ AC 係数値と 1 つの非ゼロ AC 係数がランレベルペアとしてまとめられる場合，当該非ゼロ AC 係数値が 3 ビットの符号付可変長ビット列として表されるとすると，60 ビットの信号が 19 ビットに削減されることになる．

JPEG の復号と普及

デコーダ側ではまず JFIF ファイルとしての JPEG 圧縮画像をバイト単位で走査しつつ，ECS セグメント内のハフマン符号語を検出して DC 差分値と AC ランレベルペアを抽出する．抽出された DC 差分値には逆 DPCM を，AC ランレベルペアにはランレングス復号を行い，QDCT 係数を復元後，逆量子化・逆 DCT を行った YCbCr 成分を取得する．YCbCr 成分のうち Cb・Cr 成分の内挿（補間）を行い，色差補間後の YCbCr 成分を RGB 成分に逆変換する．こうして得られた RGB 成分は，エンコーダ側の色差間引・量子化などの影響を受け，完全には復元されないものの，ほとんどの場合元と遜色ない程度に品質劣化しつつ，ファイルサイズを劇的に削減する．このシンプルなアーキテクチャと高圧縮な特性が評価され，JPEG は今なお廃れず画像符号化のデファクトスタンダードとなっている．

2.1.2 H.26x

JPEG から H.26x シリーズで加えられた改良点としては，映像の各フレームに DCT を直接適用するのではなく，直前または前後のフレームからのインター予測を行って，符号化対象の原信号を削減（時間方向に非相関化）することが挙げられる．ただし，これでは予測対象（P,B）フレーム予測の信号は削減できても，予測で用いられる参照（I）フレームの原信号が削減できない．そこで，後継の AVC ではフレーム内のブロック近傍画素から当該ブロックのピクセル値を予測するイントラ予測も用いられ，低計算量かつ高圧縮な規格として普及している．さらに，AVC ほど広く実用には至っていないものの，AVC の圧縮効率を 2 倍に向上させる HEVC が誕生した．HEVC では AVC でのイントラ予測モード数を増加させ，AVC での符号化に用いられていたコンテキスト適応可変長符号化（CAVLC）・コンテキスト適応二進算術符号化（CABAC）を CABAC 一つに限定することで，AVC よりも軽微な計算量増加で劇的な圧縮効率を図った．また，現在最新の映像符号化である VVC は HEVC の 2 倍（AVC の 4 倍）の圧縮効率を達成するために，イントラ予測モード数増加・変換ブロックサイズ増加などの新たな機能が追加されている．

AVC から採用されている映像符号化に共通のアーキテクチャを図 2.5 に示す．図からわかる通り，画像に対する JPEG 符号化とは変換・量子化という点で共通の流れをなしていることがわかる．また，この変換・量子化は映像フレームのブロックごとに行われる．この特長により，4 章で紹介の提案法は JPEG のみならず映像符号化にも共通に適用可能であることが言える．

2.2 暗号化の基礎技術

フォーマット準拠暗号化は，主に圧縮対象の信号の格納されるデータ型（符号なし／あり）やビット長，固定／可変長などのフォーマットに即しており，闇雲に暗号化を適用できるわけではない．後述の暗号化モジュールは圧縮対象の信号にも適用可能なものであり，それぞれ暗号化のセキュリティを担保

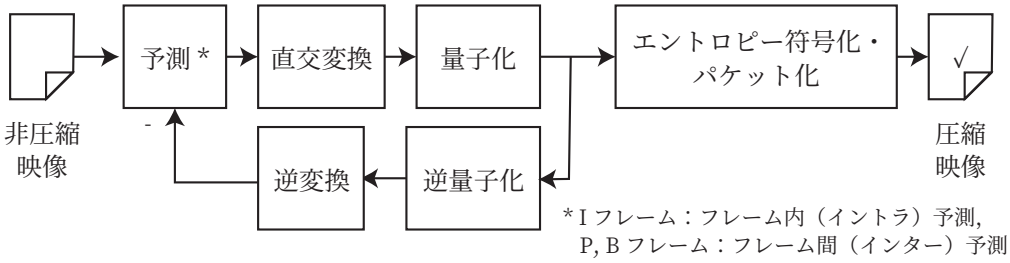


図 2.5　AVC, HEVC 映像符号化の共通フロー.

することに貢献する.

本節で示す手法のランダム性は乱数からの無作為性に依拠しており，この無作為性を達成するアルゴリズムの一つには高周期かつ等方分布性を達成する有名な擬似乱数生成器（PRNG）である MT: Mersenne twister が存在する．MT については 2.3.2 項で詳説する.

2.2.1　符号反転

符号反転とは，信号を表す係数データの符号（+/−）をランダムに反転する操作である．ある符号付サンプル $s \in \mathbb{R}$ を符号反転する操作は

$$s' = \begin{cases} -s & \text{if } \ \mathrm{rand}(i) \bmod 2 = 1 \\ s & \text{if } \ \mathrm{rand}(i) \bmod 2 = 0 \end{cases} \tag{2.7}$$

で与えられる．ここで，$\mathrm{rand}(i)$ とは i 番目サンプルに到達した時点で振られる擬似乱数である．このように，各 s について二値の乱数を振り，その値によって s の符号をランダムに反転するだけの作業であるため，極めて低計算量な暗号化となっている.

一般に，時間領域の成分であるピクセル値信号は符号無 8 ビット整数で表されることが通例であり，近年の HDR 画像フォーマット [46] や RAW 画像フォーマット [47] もダイナミックレンジの拡張はあれど符号を持つことは希であるため，この符号反転は常に適用可能なわけではない．ただし，イントラ/インター予測・DPCM といった予測後の残差信号や，DCT・離散サイン変換（DST）といった直交変換後の周波数信号などの ‘非相関化’ 後の信号は通常符号付の実数として表される．すなわち，この符号反転は非相関化領域での暗号化にしばしば用いられる．特に，周波数信号の符号反転は，周波数係数の位相 180° ランダムシフトに相当することが知られており，変換処理内の Givens 回転行列内における直接実装も可能である [23, 24].

2.2.2 排他的論理和

排他的論理和（XOR）による暗号化とは，固定長や符号ビット長の確定しているビット列をランダムに反転する操作を指す．ある符号付サンプル $s \in \mathbb{Z}$ を XOR によって暗号化する操作は

$$
\begin{aligned}
s' &= s \ \oplus \ (\mathrm{rand}(i) \ \wedge \ (2^{\mathrm{len}(s)} - 1)) \\
\mathrm{len}(s) &= \begin{cases} \lfloor \log_2 |s| \rfloor + 1 & \text{if} \ \ s \neq 0 \\ 1 & \text{if} \ \ s = 0 \end{cases}
\end{aligned}
\tag{2.8}
$$

で与えられる．ここで，$\mathrm{len}(s)$ は s のビット長を計算する関数であり，s に符号が存在しない場合 $\mathrm{len}(s) \leftarrow \mathrm{len}(s) - 1$ として扱う．また，$\wedge$ は論理積を表す．すなわち，$\mathrm{rand}(i) \ \wedge \ (2^{\mathrm{len}(s)} - 1)$ とは $\mathrm{len}(s)$ ビットのマスキング処理を $\mathrm{rand}(i)$ に施すことによって，$\mathrm{len}(s)$ ビットのランダムビット列を生成する操作に相当する．

XOR を施せる領域としては，全ての信号を 8 ビットの固定長で保持するピクセル（時間）領域や，ハフマン符号化後でビット長の保存が確定している状態でのビットストリーム領域などがある．一方で，ビット長の保存がされない QDCT 係数領域で可変長のランダムビット列を XOR 演算に用いた場合，XOR 対象の s の最高位ビットが 0 になることがあるため，暗号化ビット列のビット長が変化してしまうことで，同じランダムビット列の取得とそれによる平文化が不可能になる．そのため，ビット長の取り出されていない可変長ビット列に XOR を適用する場合，この平文化を可能にするために元のビット長をサイド情報として保存しなければならない．このサイド情報はコンテンツの解像度の増大に伴って増加してしまうため，本研究ではサイド情報を増加させないために，単純にサンプルのビット長保存が確定している（サイド情報の不要な）状況での暗号化を想定する．

2.2.3 シャッフル

シャッフルは，n 個サンプルの集合 $\mathbf{s} := \{s_i\}_{i=1}^{n}$ を行ベクトル，置換行列を $\mathbf{P}_n$ としたとき，

$$
\mathbf{s}' = \mathbf{s}\mathbf{P}_n \tag{2.9}
$$

と表される．$\mathbf{P}_n$ を作成するアルゴリズムは種々存在するが，メモリ効率と優れた分布特性の観点から，古くから Fisher-Yates や Durstenfeld などの手法 [48] が知られている．

2.3 乱数生成

暗号化を実現するためには，既存のある慣習に基づいたアルゴリズム的操作をランダムに見せる必要がある．このランダム性を利用するためには，暗号化キーからの乱数生成を利用して，その乱数列を暗号化アルゴリズムに用いることが一般的であり，代表的な乱数生成として従来では ‘カオス写像’・‘乱数生成器’ という二通りのメカニズムを駆使する．前者は初期状態を暗号化キーから適切にマネージしな

くてはならないという煩雑なアルゴリズムになるため，本研究ではこの初期状態を暗号化キーから効率的にマネージできる後者のメカニズムを利用することにし，そのメカニズムについて説明する．

2.3.1 乱数生成器

乱数生成器（RNG: random number generator）とは，あるランダムな数列を生成するアルゴリズムをいう．ここでは，2.3.2 項の ‘擬似乱数生成器’ と区別するために，どの初期状態からでもランダムな数列を生成するものを説明する．あるランダムな数列を作成したとすると，これらの数列には何の作為性もない．そのような無作為なサンプルは，ハードウェアからの熱雑音などから物理的に生成できることが報告されている [49].

このように ‘どの状態からでもランダム’ な数列であれば，ある時点での乱数を仮に取得されたとしても，それ以降の乱数を推定・利用されることがないため，高安全といえる．ただし，それはすなわち，ある初期状態を用いたとしても ‘それ以降の状態が再現できない’ ことを意味し，同じ乱数列の再現には元と同一の乱数列を全て保存しておかなければならず，暗号化技術に用いるものとしては非効率である．ある状態から次の状態を再現できるものとして擬似乱数生成器が存在する．

2.3.2 擬似乱数生成器

擬似乱数生成器（PRNG: pseudo RNG）とは，ある初期状態（ベクトル）を利用してそれ以降の擬似的な乱数列を生成するアルゴリズムである．擬似的であれば，ある共通の初期状態となるベクトルからそれ以降のベクトルの列を同じ生成順序で生成でき，実際の暗号化に用いた乱数列を全て保存しておかずとも，元と同一の初期状態ベクトル（ないしその初期化に用いた暗号化キー）一つのみ保存しておけば良いことになる．

古典的な PRNG には C 言語の線形合同法を用いた `rand()` 関数がある．また，現在最もポピュラーかつ実用的な PRNG には MT: Mersenne twister [50] があり，Python の `random` モジュールや Matlab の `rng()` 関数などで手軽に用いられている．このように様々な PRNG が存在する一つの理由には乱数の周期がある．C 言語の `rand()` 関数では各乱数はその内部状態である 32 ビットベクトルから直接生成されるため，$2^{32}-1$ 回乱数を生成すれば，ある初期状態ベクトルと同じベクトルが再び出現し，攻撃者は現実時間における状態ベクトル空間の総当たりでいかなる乱数も取得できてしまう．しかし，MT では 19936 ビットの incomplete array を内部状態ベクトルとして持ち，乱数はそのベクトルから調律行列を用いて所望の精度に（ランダム性を保持しつつ）変換・出力（生成）されるため，周期は $2^{19937}-1$ と膨大である．すなわち，攻撃者はある乱数の取得に $2^{19937}-1$ 回の総当たりを行う必要があり，そのような総当たりは現実的に不可能とされている．

また，MT は線形回帰に基づくため，ある乱数が取得されてしまうとそれ以降の乱数生成が予測できてしまい ‘暗号学的に安全’ ではないとされており，近年では暗号学的に安全な PRNG（CSPRNG:

cryptographically secure PRNG）の開発も世界中で進められている [51]．ただし，ある乱数が取得されることが安全でないとされるのは，その乱数がとあるアルゴリズムの特定部分に適用されることが明白である場合であり，本研究での暗号化アルゴリズムでは適用中の乱数列は保存されず，また状態ベクトルを初期化するための暗号化キーもアルゴリズム中に埋め込まない場合を想定している．すなわち，フォーマット準拠暗号化への PRNG 適用という観点では MT は‘安全ではないとしても脆弱ではない’といえるため，本研究では簡単のため MT を暗号化アルゴリズムに適用する．

上述のように，MT は 19936 ビットの状態ベクトルを内部状態として持つが，その初期状態は 19936 ビット未満で充分に安全な長さであれば任意長のベクトルで設定できる．このため，本研究では SHA-2 アルゴリズム [52] で生成した 256 ビットのダイジェストを暗号化キーとして MT の初期状態に設定（入力）する．256 ビット SHA-2 ダイジェストで初期化された MT から生成される乱数列は，符号反転や排他的論理和での rand(i)，シャッフルでのランダム置換生成などのほか，全体のうちどの割合を暗号化するかの確率などに用いられ，これらに用いられた以降のどの場面にも残らない．よって，使用中・後の乱数列を攻撃者に知られる・推定されることもない．

2.4 暗号文単独攻撃

フォーマット準拠暗号化は通常，共通鍵暗号系での適用を前提としており，キーを安全に管理すれば既知平文攻撃・選択平文攻撃・選択暗号文攻撃[*1]などの攻撃は通用しないものと考えられている．一方，キーの考慮を前提としない暗号文単独攻撃（COA: ciphertext-only attack）については，暗号文（フォーマット準拠暗号化の場合，暗号化された信号ないしそれによる画像）から平文（暗号化なしの信号・画像）を復元・近似される恐れがあるため，様々な COA の可能性が報告されている．本章ではそれらの COA について説明する．

2.4.1 アルゴリズム総当たり攻撃

アルゴリズム総当たり攻撃とは，あるアルゴリズムが平文から生み出しうる暗号文の数（暗号化の結果数）を考察し，それを攻撃に用いることで平文そのものを復元しようとする攻撃である．一般に現在，暗号化のキーとして用いる数列の長さとして 256 ビット以上を用いると，2030 年までそのキーのビット列を総当たりできないことがアメリカ国立標準技術研究所（NIST: National Institute of Standards and Technology）により報告されており [53]，この 256 ビットという長さを基にして考えると，256 ビットの総当たり数 2^{256} よりもアルゴリズム総当たり攻撃が簡単であれば，攻撃者によって好都合となってしまう．そのため，キーの総当たりよりも複雑な総当たりを要する暗号化の設計が求められる．

[*1] それぞれ，暗号文から平文の一部が分かる場合にキーを推測する攻撃，大量の平文から暗号文のペアを入手できる条件で解読対象の平文から暗号文を推測する攻撃，解読対象以外の暗号文から平文を大量に入手できる場合で暗号文から平文を推測する攻撃．

(a) (b) (c)

図 2.6　ジグソーパズル解読攻撃：(a) 元画像 *ucid00231*, (b) (a) を 16×16 ブロック単位でシャッフルした画像, (c) (b) をジグソーパズル解読攻撃で 89.5 % 再構成した画像.

2.4.2　ジグソーパズル解読攻撃

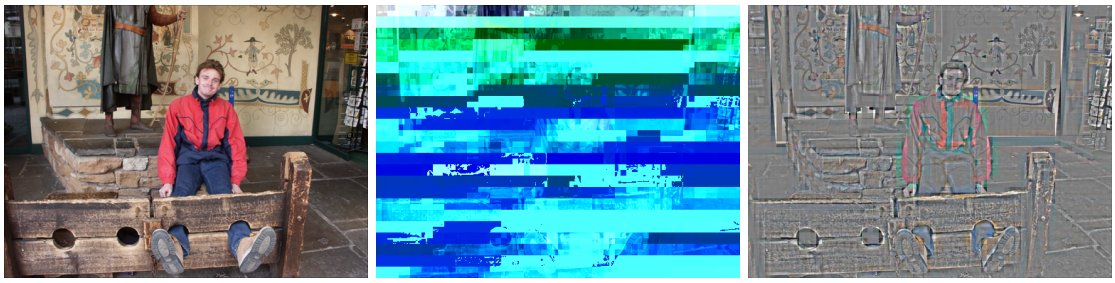
ジグソーパズル解読攻撃とは，暗号化画像のなかに元画像のテクスチャが暗号化されず残っている場合，これをつなぎ合わせて元画像を復元しようとする攻撃である．たとえば，ピクセルブロックの入替による暗号化画像を正方形ピースからなるパズルに見立て，ピース同士を結合させることで元画像の復元を図る（図 2.6）[54]．一般に，非相関化された中で暗号化された信号は，デコード時に ‘相関化’ されてある範囲ないし全体に及んで元画像のテクスチャを秘匿化するため，このような攻撃は通用しない．ただし，画像をブロックに分割してブロック内はほとんど暗号化しない暗号化 [27] だと，このジグソーパズル解読攻撃によって破られる危険性を孕んでいるため，ジグソーパズル解読攻撃に対する頑健性を検証する必要がある．

2.4.3　置換攻撃

置換攻撃とは，ある信号成分だけ暗号化してそれ以外を暗号化しないような場合，暗号化成分を別の値に置換することで，非暗号化部分を漏洩しようとする攻撃である．たとえば，JPEG 符号化中の非相関化された周波数領域で DC 係数のみ暗号化した場合，暗号化された DC 係数値をゼロに置換することで，非暗号化部分である AC 係数の情報を漏洩させようとする ‘DC 係数削除’ が報告されている [55]（図 2.7）．このような攻撃に対抗するため，置換対象の信号をできる限り暗号化できるようなフォーマット準拠暗号化の設計（たとえば信号成分の位置関係の変更）が求められる．

2.4.4　スケッチ攻撃

スケッチ攻撃とは，暗号化後の信号成分の大まかな位置関係が保存されていることを考慮し，暗号化画像からそのラフスケッチを取得しようとする攻撃である．置換攻撃でも各信号成分の位置関係が不変の場合に適用可能であるが，このスケッチ攻撃はより大きな括りの信号集合をまとめて一つの値に置換する．たとえば，JPEG 符号化中で 8×8 QDCT 係数ブロックごとにその内部を暗号化するようなア



(a) (b) (c)

図 2.7　置換攻撃：(a) 元画像 *ucid00231*, (b) (a) を JPEG 符号化の DPCM 後で，DC 差分値を暗号化した圧縮画像（$Q = 70$）, (c) (b) の DC 差分値を置換攻撃で全てゼロに置換した画像.

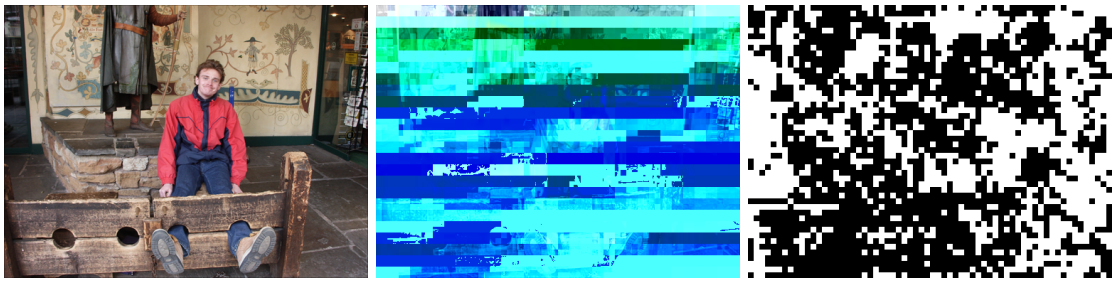
ルゴリズムに対し，置換攻撃では暗号化された部分を一つ一つ置換するが，スケッチ攻撃では 8×8 ブロック内の（たとえば）‘非ゼロ係数個数’ をまとめて一つの値に置換する．具体的には，ある 8×8 ブロック内の非ゼロ係数個数が範囲 $[r_1\ r_2]$ $(r_{1,2} \in [0\ 64];\ r_1 \neq r_2)$ 内に当てはまるとき，当該ブロックの 8×8 ピクセル値を 0 とし，それ以外のブロックのピクセル値を 255 に当てはめる（あるいは，白黒を逆としても良い）非ゼロ係数カウント（NZCC: nonzero coefficient count） [43] が存在する（図 2.8）．NZCC はブロック内の AC 係数をまとめてブロック間で入れ替える手法で容易に対策可能であるが，DC 差分値の絶対値のみ残っていた場合でも濃淡を描画できるような変種や，各ブロック内の非ゼロ AC 係数の個数のみをカウントする変種なども存在する．このような攻撃に対抗するため，QDCT 係数領域の DC 係数・AC 係数を共に元のブロックから別々かつランダムに入れ替えるような暗号化 [39] がすでに提案されている．

2.5　従来のフォーマット準拠暗号化

これまで紹介した基礎技術を用いて，多くの画像映像フォーマット準拠暗号化が提案されている．以下に，いくつかの手法を概説する．ここで，JPEG フォーマット準拠暗号化において，JPEG 符号化前のピクセル信号に施すフォーマット準拠暗号化を ‘JPEG 符号化前フォーマット準拠暗号化’，JPEG エントロピー符号化の DPCM・RLE 前で施すフォーマット準拠暗号化を ‘JPEG 符号化内フォーマット準拠暗号化’，DPCM・RLE 後で施すフォーマット準拠暗号化を ‘JPEG 符号化後フォーマット準拠暗号化’ として分類する．

ピクセルブロックベース暗号化

ピクセルブロックベース暗号化 [27] は JPEG 符号化前フォーマット準拠暗号化である．この手法は画像のピクセル信号を 16×16 のブロックごとで跨がないように暗号化することで，JPEG 符号化効率を極力阻害しない上，再符号化にも対応している．この手法では，まず画像を $16i \times 16j$ $(i, j \in \mathbb{N})$ ピクセルブロックに分割し，四つの暗号化モジュール：ブロック回転／反転，ブロック入替，ブロック



(a) (b) (c)

図 2.8　スケッチ攻撃：(a) 元画像 *ucid00231*, (b) (a) を JPEG 符号化の DPCM 後で，DC 差分値を暗号化した圧縮画像 ($Q = 70$) , (c) (b) のアウトラインをスケッチ攻撃 NZCC ($[r_1\ r_2] = [15\ 25]$) で描画した画像.

ネガポジ反転，ブロック色成分入替を適用する．ブロック回転／反転では，ブロックを時計回り（ないし反時計回り）に $0°, 90°, 180°, 270°$ のいずれかでランダムに回転し，垂直・水平方向でブロックをランダムに反転する．ブロック入替では，ブロック同士をランダムに入れ替える（シャッフルする）．ブロックネガポジ反転では，ブロックの色をランダムに反転する．ブロック色成分入替ではブロック内の RGB 色成分順序をランダムに入れ替える．この手法では暗号化画像を JPEG 符号化して SNS にアップロードした後，その JPEG 圧縮画像を SNS 側で自動的に再符号化（コンテンツを一度全て復号し，再び JPEG 符号化）した後でも，その再符号化画像をデコード・平文化しても，暗号化なしの場合に比べ品質劣化がほとんどないという点で優れている．しかし，3.2 章で説明するように，この従来法では暗号化画像のブロック内の模様をほとんど暗号化できず，ジグソーパズル解読攻撃に対する根本的脆弱性を拭えないという欠点もある．

周波数係数のランダム符号反転

ランダム符号反転（RSF: random sign flip, 2.2.1 項）は主に DCT 係数・QDCT 係数の符号をランダムに反転する手法，つまり JPEG 符号化内フォーマット準拠暗号化の手法として有名である [23, 24]. これらの手法では，周波数係数を極めて低計算量の処理で暗号化し，模様情報も秘匿化できるものの，周波数係数の位置が不変であることから置換攻撃への耐性を持たない．また，各係数の符号を単純にランダム反転するのみであるため，知覚劣化度合の調整可能性を持たない.

ジグザグスキャンランダム化

ジグザグスキャンランダム化（RANDZZ: randomized zig-zag scan） [22] は 8×8 QDCT 係数ブロック内をシャッフルすることによって，本来ジグザグスキャンで走査するブロックをランダムな順序でスキャンする JPEG 符号化内フォーマット準拠暗号化の手法である．8×8 ブロック内の QDCT 係数における位置関係が秘匿化されるため，総当たり攻撃・置換攻撃には頑健であるものの，最適なジグザグスキャン順序を行わないために JPEG 圧縮画像のビットレートを甚だしく増加させ，符号化効率

を阻害する．また，8×8 全係数の入替であるために知覚劣化度合の調整可能性を持たない．

ビットプレーン内入替

ビットプレーン内入替（IBS: intra-bitplane shuffling） [42] は量子化後の符号付十進数（周波数係数）をビットプレーンに分解し，各ビットプレーン内ごとのビットをランダムに入れ替える JPEG 符号化内フォーマット準拠暗号化の手法である．ビット単位での入替であるため，RANDZZ のように総当たり攻撃・置換攻撃に頑健であり，設計次第では知覚劣化度合の適度な調整可能性を持たせられるものの，符号化効率は甚だしく阻害される．

全ブロック間入替

全ブロック間入替（FIBS: full inter-block shuffle） [43] は QDCT 係数ブロック間で，同帯域の QDCT 係数をランダムに入れ替える JPEG 符号化内フォーマット準拠暗号化の手法である．同帯域間での係数の入替であるため JPEG 圧縮画像のビットレート増加（符号化効率阻害）を 20 % 程度にまで抑えつつ，総当たり攻撃・置換攻撃・スケッチ攻撃に強い耐性を持つが，64 帯域の QDCT 係数の入替であるために知覚劣化度合の柔軟な調整可能性を持たない．

DC ビットプレーン・AC 係数入替

DC ビットプレーン・AC 係数入替（DCACS: DC bitplane and AC coefficient shuffling） [44] は DC 係数をビットプレーンに分解した際の LSB プレーンから 7 プレーンに存在する各ビットを DC 係数間でランダムに入れ替え，また各ブロック内の上位 5 帯域の AC 係数をブロック内外でランダムに入れ替える JPEG 符号化内フォーマット準拠暗号化の手法である．DC 係数・AC 係数ともに入替の範囲を制限することで，DCACS は JPEG 符号化効率を維持し適度な知覚劣化度合の暗号化画像を生成するが，コンテンツの一部のみの暗号化であるためにその部分以外の非暗号化部分を置換攻撃で漏洩されやすい．なお，DC ビットプレーンの暗号化数・AC 係数の暗号化帯域数などは調整可能であるため，知覚劣化度合の潜在的な調整可能性を持つ．

ランレベルペア入替

ランレベルペア入替（RPS: run-level pair scrambling） [38] は RLE 後の AC 係数ランレベルペア（ある当該の非ゼロ AC 係数前に連続するゼロのランレングスと，その非ゼロ AC 係数からなるペア）を各ブロック内でシャッフルする JPEG 符号化後フォーマット準拠暗号化である．符号化後の暗号化のため，JPEG 圧縮画像のビットレートを増加させず置換攻撃にも頑健である．ただし，各ブロックは元の位置から変わっていないために依然としてスケッチ攻撃によってブロックごとのアウトライン情報を描画されやすいということに変わりはない．ブロック入替を追加した RPS [39] では，元ブロックの位置が他ブロックと入れ替わるためにスケッチによるアウトライン描画すら困難となっている．また，従

来のブロック内 RPS [38] が各ブロック内でしかランレベルペアをシャッフルできないという問題に対し，ブロック間 RPS [41] はブロック間でランレベルペアをシャッフルするが，シャッフル後ランレベルペアを各ブロック内に配置する際に，配置不全が起こり平文化できないという危険性を孕んでいる．

符号ビット列 XOR

JFIF ECS セグメント内ビット列の XOR [40] は，ハフマン符号化で符号化された DC 差分値・非ゼロ AC 係数のビット長に従って，JFIF ファイル ECS セグメント内のビット列をランダムに反転する JPEG 符号化後フォーマット準拠暗号化の手法である．ただし，ECS セグメント内ビット列に XOR を施せるのは，その対象ビット列のビット長が明にビット長保存される場合であり，ビット長保存しない符号化を用いる場合には暗号化ビット列を平文化するための DC 差分値・非ゼロ AC 係数ビット長をサイド情報として保存する必要があり，JPEG 以外の符号化には適していない．

2.6 固定長符号付二進化

二進領域内の部分的な暗号化 [42, 44] は，暗号化デコード画像の知覚劣化度合に（潜在的に）適度の調整可能性を持たせられる．十進数を二進領域に変換する二進化として従来よく用いられてきたのは，D ビットの固定長符号付二進化（FSB: fixed-length signed binarization）である．この二進化は符号付十進数を c，n ビットの固定長二値列を $\mathbf{b}$，ある符号付十進数を x，二進数を $b_i \in \{0, 1\}$，符号関数を sgn()，絶対値関数を abs() とすると，

$$\mathbf{b} = \mathrm{sgn}(c)|(\mathrm{abs}(c))_{2,n-1} \tag{2.10}$$

$$\mathrm{sgn}(x) = \begin{cases} 0 & \text{if } x \geq 0 \\ 1 & \text{otherwise} \end{cases} \tag{2.11}$$

$$\mathrm{abs}(x) = \begin{cases} x & \text{if } x \geq 0 \\ -x & \text{otherwise} \end{cases} \tag{2.12}$$

$$(x)_{2,\eta} = b_{\eta-1} \cdots b_0 \tag{2.13}$$

と表せる．この二進化は JPEG2000 EBCOT [56], HEVC CABAC [57] などの主要なエントロピー符号化部で用いられている．ここで，(2.13) はお馴染みの十進-二進変換であり，十進数 x の η ビット二値列への変換を意味する．

第 3 章

ピクセルキューボイドベース暗号化

本章では，映像フレームごとの JPEG 符号化を行う映像符号化規格 MJPEG のためのピクセルキューブベース暗号化 [35, 58] を提案する．ピクセルキューブベース暗号化は，再符号化可能性を実現する従来のピクセルブロックベース暗号化 [27] の抱えるジグソーパズル解読攻撃への潜在的な脆弱性を撤廃することでその再符号可能性を担保する．また，ピクセルブロックベース暗号化よりも符号化効率を阻害しうる可能性をもつ代わりに，ピクセルブロックベース暗号化よりも攻撃耐性を向上させる．さらに，ピクセルキューブベース暗号化に内在するキューブ状ジグソーパズル解読攻撃（後述）への脆弱性をも撤廃するために，映像をランダムな辺幅の直方体（キューボイド）で暗号化するピクセルキューボイドベース暗号化 [36] へと拡張する．

3.1 動機付け

2.5 節で紹介した従来のピクセルブロックベース暗号化 [27] は，MJPEG で符号化される映像フレームを $16i \times 16j$ $(i, j \in \mathbb{N})$ ブロックに分割し，ブロック回転反転・ブロック入替・ブロックネガポジ反転・ブロック色成分入替の四モジュールからなる暗号化を適用する．暗号化後のブロックが MJPEG の符号化ブロックを跨がないために，ピクセルブロックベース暗号化では MJPEG 符号化効率を維持できるという性質を持つ．しかしながら，暗号化フレームの各ブロック内はほぼ暗号化されず，元フレームブロック内の模様情報が残るために，暗号化フレームを完全に／ほとんど再構成されてしまうという脆弱性を撤廃できない．この攻撃への頑健性は上述のブロック入替などの暗号化モジュールを複雑に組み合わせ，ブロックの大きさを（最小 16×16 ピクセルとして）小さくすればするほど高められると述べられているものの，元画像のブロックが残存している限り，その頑健性は常に脅かされる．この頑健性をさらに強化しつつ，再符号化可能性も維持するための一解法として，映像フレームをピクセル（時間）領域で暗号化しつつ，元フレームにないブロックを元映像のフレームを束ねた時空間平面から出現させることが考えられる．ブロック内の秘匿性を向上させて単一フレーム内での不正な復元を防止する

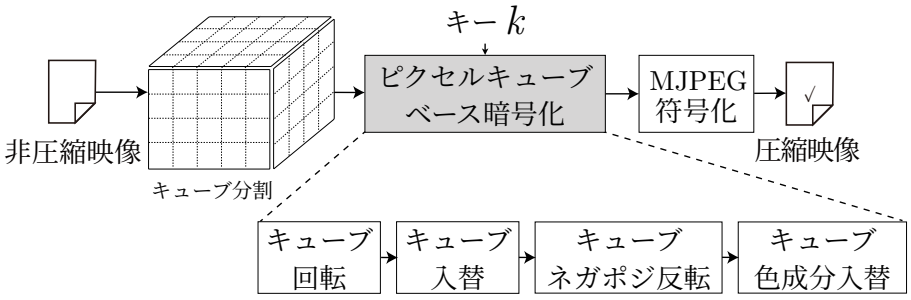


図 3.1　ピクセルキューブベース暗号化の暗号化アルゴリズム（フロー）.

表 3.1　キューブ回転の各方向における回転角のバリエーション・乱数との対応例.

乱数	0	1	2	3
回転角	0°	90°	180°	270°

ために，本章ではピクセルキューブベース暗号化・ピクセルキューボイドベース暗号化を提案する.

3.2　ピクセルキューブベース暗号化

提案法の暗号化アルゴリズムを各モジュールに分けて構成したフローを図 3.1 に示す．図 3.1 から分かる通り，提案法は MJPEG 符号化前の映像フレームをまとめて一つの巨大な直方体（キューボイド）に見立てて立方体（キューブ）に分割し，四つのモジュール（キューブ回転，キューブ入替，キューブネガポジ反転，キューブ色成分入替）からなる暗号化をに施す．以下にその四つのモジュールの詳細を示す.

キューブ回転

キューブ回転は，各キューブを垂直・水平・奥行方向にそれぞれ（$0°, 90°, 180°, 270°$ の中から）ランダムに回転するモジュールである[*1]（図 3.2）．回転のバリエーションは，表 3.1 に示すように $0°, 90°, 180°, 270°$ の 4 通りが存在し，各回転に 0 から 3 と番号付けしておき，その対応を用いて垂直・水平・奥行方向にランダムに回転する．奥行方向への回転角が 90° か 270° である場合，回転対象のキューブのフレーム上に向いている面が，回転後に映像の時間方向の断面と置き換わる．これにより，元フレーム内に本来存在しないブロック（時間方向の面）が暗号化後に出現し，単一フレームに対する攻撃を防止できる．この時間方向の面は画像が本来満たしている ‘隣接画素同士が滑らかな変化である’ という性質を満たさないことがあり，ブロックのサイズを適切に選んだとしても MJPEG 符号化効率に影響を及ぼす．ただし，回転角を制限することでこの影響は抑えられる.

[*1] たとえば，奥行方向に 90°，垂直方向に 270°，奥行方向に 180° 回転する組合せなど

表 3.2　キューブ色成分入替における，入替後の色成分順序の例.

番号	入替後の RGB 色成分順序
0	R→G→B
1	R→B→B
2	G→R→B
3	G→B→R
4	B→R→G
5	B→G→R

キューブ入替

キューブ入替は各キューブの位置をランダムに入れ替えるモジュールである（図 3.3）．全体を N 個とすると 0 から $N-1$ までの番号を入替対象のキューブに順番に索引付けしておき，N 個の索引をシャッフルした後，その索引集合からランダムに 2 個ずつ選んで対応するキューブ同士を入れ替える．この作業を，未処理のキューブが無くなるまで繰り返す．このキューブ入替は元フレーム内に存在するブロックだけで実行されるため，MJPEG 符号化効率にほとんど影響を及ぼさない．

キューブネガポジ反転

キューブネガポジ反転はキューブの色をランダムに反転するモジュールである．JPEG では，あるキューブ内の RGB ピクセル値が 0 から 255 の値をとるため，それらの値を最大ピクセル値の 255 からランダムに減算する．この操作も元フレーム内に存在するブロックだけで実行されるため，また反転後のピクセル値も反転前と同様に画像の平滑性を満たすため，MJPEG 符号化効率にほとんど影響を及ぼさない．

キューブ色成分入替

キューブ色成分入替はキューブ内の RGB 色成分順序をランダムに入れ替えるモジュールである．並べ替えのバリエーションは，表 3.2 に示すように 6 通りが存在し，各入替に 0 から 5 と番号付けしておき，その対応を用いてキューブ内の色成分順序をランダムに並べ替える．

3.2.1　セキュリティ

ここでは，キューブベース暗号化の各モジュールに対してありうる攻撃手法とその耐性を議論する．キューブベース暗号化に対して現状考えうる攻撃は，アルゴリズム総当たり攻撃とジグソーパズル構成攻撃がある．

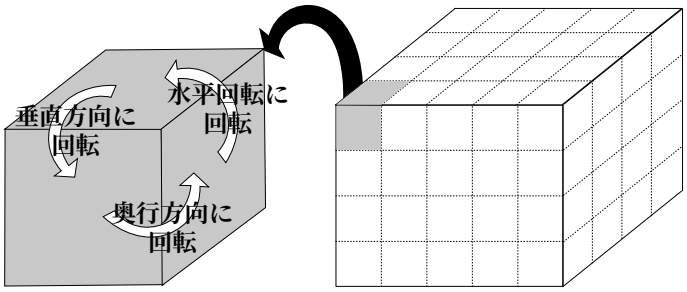


図 3.2　キューブ回転.

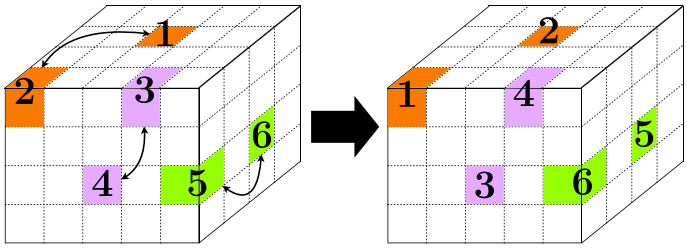


図 3.3　キューブ入替.

アルゴリズム総当たり攻撃に対する頑健性

まず，アルゴリズム総当たり攻撃に対する耐性を考察する．キューブ回転では，一つのキューブを奥行・垂直・水平の三方向で，それぞれ $0°, 90°, 180°, 270°$ の四つの中からランダムな回転角で回転するため，これを偶然正解の位置に逆回転する確率は $1/4^3$，それが N 個同時で $(1/4^3)^N = 1/64^N$ の確率となる．すなわち，N 個のキューブ回転を復元するための総当たり数は

$$N_{CR} = 64^N = 2^{6N} \tag{3.1}$$

となる．また，キューブ入替に対しては，入替後の N 個キューブをランダムに入れ替えつつ，その正解パターンを総当たりすることが考えられる．すなわち，N 個中から一つを正解の位置に配置する確率が $\frac{1}{N}$ であるため，その後さらにもう一つも正解の位置に配置する条件付確率は $\frac{1}{N(N-1)}$ と考えられ，N 個キューブを偶然正解の位置に配置する条件付確率は $\frac{1}{N!}$ となる．この確率を 1 にするための総当たり数は $N!$ であることから，キューブ入替の総当たり数は

$$N_{CS} = 2^{\log_2 N!} = 2^{\sum_{i=1}^{N} \log_2 i} \tag{3.2}$$

となる．キューブネガポジ反転に対しては，N 個のキューボイドの色を全て正しい色に反転し直される条件付き確率を $1/2^N$ と求められる．したがって，N 個のキューブネガポジ反転を復元するための総当たり数は

$$N_{CN} = 2^N \tag{3.3}$$

となる．キューブ色成分入替に対しては，N 個のキューボイドの色成分を全て正しい配置に逆入替し直される条件付き確率を $1/6^N$ と求められる．したがって，N 個のキューブ色成分入替を復元するための

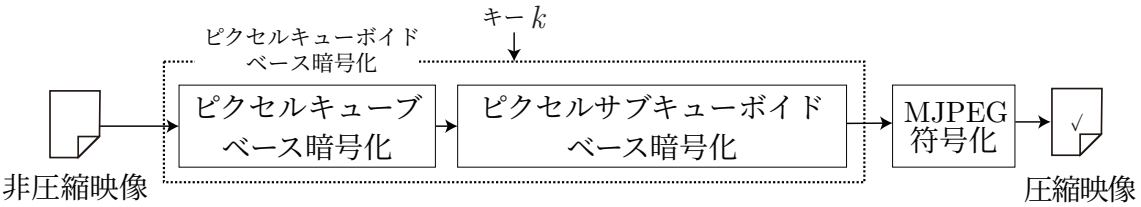


図 3.4　ピクセルキューボイドベース暗号化の暗号化アルゴリズム（フロー）.

総当たり数は

$$N_{CC} = 6^N = 2^{N \log_2 6} \tag{3.4}$$

となる．よって，ピクセルキューブベース暗号化の全モジュールの施された暗号化映像を復元するために充分な総当たり数は

$$N_C = N_{CR} N_{CS} N_{CN} N_{CC} = 2^{7N + N \log_2 6 + \sum_{i=1}^{N} \log_2 i} \tag{3.5}$$

と求められる．ここで，256 ビットの SHA: secure hash algorithm-2 ハッシュ値を暗号化のキーとして扱うことを安全性の基準として考慮すると，256 ビット SHA-2 ハッシュ値（キー）一つを探索するために充分となる総当たり数は 2^{256} 回であるため，アルゴリズム総当たり数はこの数よりも小さくならない場合に，キーの総当たりよりも簡単にならないといえる．つまり，$2^{7N + N \log_2 6 + \sum_{i=1}^{N} \log_2 i} \geq 2^{256}$ となる最小のキューブ数 N で暗号化対象の映像をキューブ分割すれば，ピクセルキューブベース暗号化へのアルゴリズム総当たりに対する頑健性は担保される．この式より，$N \geq 21$ であればピクセルキューブベース暗号化はアルゴリズム総当たりに対し頑健であり，非常に少ないキューブ数で分割しても安全であることがいえる．

ジグソーパズル解読攻撃に対する頑健性

また，ジグソーパズル解読攻撃への頑健性を考察する．ここでは，映像フレーム内のブロックをジグソーパズルピースに見立てて結合する解読を対象に考える．Chuman らは，ピクセルブロックベース暗号化におけるブロックの個数が充分に大きく，多くの暗号化モジュールを組み合わせる場合，この攻撃に充分頑健であることを示している [59]．ピクセルブロックベース暗号化のブロック回転反転で出現させられるブロックのバリエーションは提案法のキューブ回転で出現させられるバリエーションに包含されており，なおかつキューブ回転ではブロック回転反転で出現させられない時間平面・他フレームブロックを元フレームに出現させられるため，提案法はピクセルブロックベース暗号化よりもジグソーパズル解読攻撃に頑健であることがいえる．ただし，ブロックベースのジグソーパズル解読攻撃が提案法に無効であっても，キューブベースのジグソーパズル解読攻撃が有効である可能性があるため，本研究ではこの攻撃への耐性も考察する．ブロックベースジグソーパズル解読攻撃では，暗号化画像（フレーム）のブロック辺同士をマッチングすることで，正解の画像（映像フレーム）一枚を復元しようとした．

表 3.3　実験で使用した入力映像シーケンスとその仕様．動作部分・停止部分とはそれぞれ，映像中で時間方向に変化した領域の大きさを主観的にまとめたもの．

入力映像	*Akiyo*	*Bowing*	*Coastguard*
動作部分	小	中	大
停止部分	大	中	小
サイズ	垂直 288 × 水平 352 × 奥行 256		
色深度	8-bit RGB 3 成分		

これに対し，キューブベースジグソーパズル解読攻撃では，暗号化映像全体のキューブ面同士をマッチングすることで，正解の映像全体を復元することが予想される．このキューブ面同士は，互いに同じ大きさであり，JPEG 圧縮歪みを含んでいるという状況である．これらは，開示されている限り攻撃者の知りうる情報であり，従来のブロックベースジグソーパズル解読攻撃を拡張することで明らかに試行される ‘可能性を拭うことはできない’．そこで，図 3.4 のピクセルサブキューボイドベース暗号化が有効な防御手段となる．このピクセルサブキューボイドベース暗号化は，ピクセルキューブベース暗号化の各モジュールをランダムな辺幅の直方体（キューボイド）によって行う手法へ拡張したものであり，ピクセルキューブベース暗号化で全体的に暗号化された映像のランダムな一部を，奥行方向にランダムな辺幅のサブキューボイドで暗号化する．この拡張版を加えた全体としてのピクセルキューボイドベース暗号化では，ピクセルキューブベース暗号化による暗号化映像の構成するキューブを（ランダムな辺幅の）キューボイドに変化させるため，攻撃者にキューブの大きさの情報すら与えない．すなわち，攻撃者は映像中のサブキューボイドの大きさを検出／推定することから始めなければならず，その検出／推定が「正解かもしれない」という不確かさの上に攻撃を試行することになり，統一な大きさのキューブを用いたキューブベースジグソーパズル解読攻撃はもはや功を奏さなくなる．この拡張版については 3.4 章で詳述する．

3.3　実験

提案法の有効性を示すための実験を, 以下の条件と手順で行なった．テスト映像には *Akiyo*・*Bowing*・*Coastguard* [60] を用いた．映像シーケンスの仕様を表 3.3 にまとめる．各入力映像シーケンスで時間方向に変化した（動作）部分が大きい場合，その部分がキューブ回転などでフレーム内に出現することによって，MJPEG での圧縮効率に影響が及ぶ．そのため，本研究では動作部分の異なる映像に対して提案法を施した際の，暗号化映像フレームの主観評価と圧縮効率の比較を行なった．実験の手順は以下となる：

1. 入力映像を暗号化する．

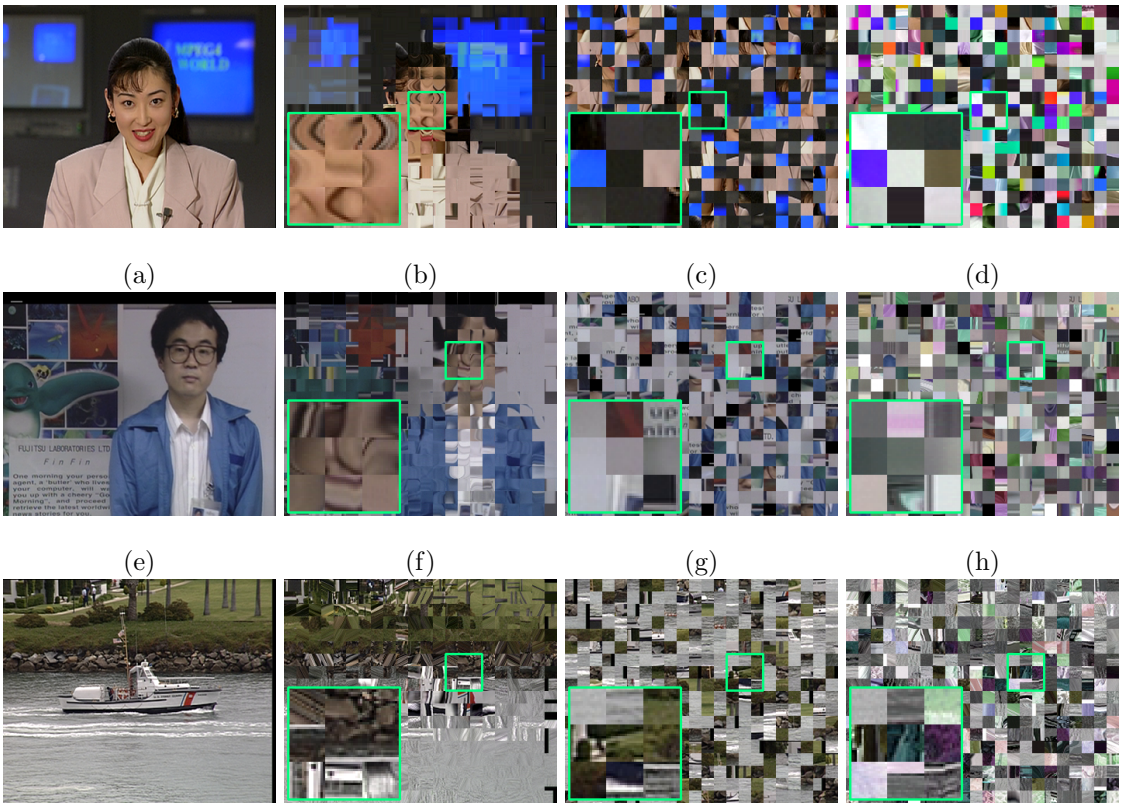
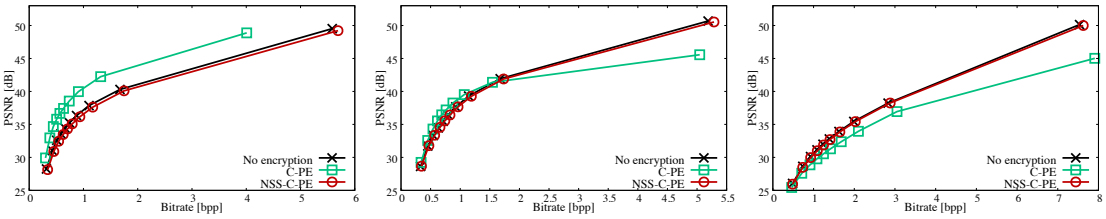


図 3.5　入力映像の提案法での暗号化結果：（上行から下行に）*Akiyo*200 番目フレーム，*Bowing*160 番目フレーム，*Coastguard*140 番目フレーム，（左列から右列に）元映像フレーム，キューブ回転適用結果，キューブ入替適用結果，ピクセルキューブベース暗号化全モジュール適用結果．

2. 暗号化フレームを MJPEG 符号化（品質ファクタ $Q = 10, 20, \cdots, 100$）する．
3. 圧縮フレームの平均ビットレートを計算する．
4. 圧縮フレームを MJPEG 復号する．
5. 復号フレームを平文化する．
6. 入力フレームと平文化フレームの平均 PSNR を計算する．

提案法での暗号化映像フレームを図 3.5 に示す．図 3.5(b,f,j) から，キューブ回転のみ施した場合でも，暗号化後の単一フレームのみを扱うブロックベースのジグソーパズル解読攻撃が無意味であることがわかる．キューブ入替のみを施した場合，時間平面からではなく他フレームからのブロックを元フレーム内ブロックと入れ替えるため，どのフレームかに含まれているブロックは元フレーム内に出現していることが推測できるものの，明らかに元フレーム内のブロックは他フレームのものと置き換わっている（図 3.5(c,g,k)）．ピクセルキューブベース暗号化全モジュールの適用により，時間平面・他フレームのブロックと置き換わった元フレーム内ブロックの色までもが暗号化されるため，元フレームのブロックの推測がもはや困難となる（図 3.5(d,h,l)）．



(a) (b) (c)

図 3.6　レート歪み曲線を用いた提案法での暗号化映像フレームの MJPEG 圧縮効率の比較：(a)*Akiyo*，(b)*Bowing*，(c)*Coastguard*.

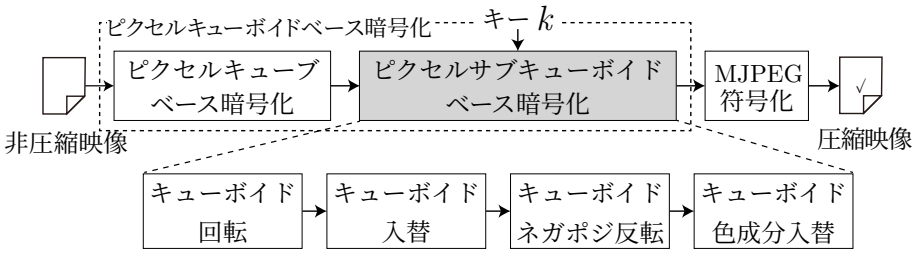


図 3.7　ピクセルサブキューボイドベース暗号化のフロー.

また，提案法での暗号化映像フレームの MJPEG における圧縮効率を図 3.6 に示す．図 3.6 での‘C-PE’（緑線）は，ピクセルキューブベース暗号化全 4 モジュールでの暗号化映像フレームを MJPEG で符復号化した場合の圧縮効率を示している．そして，‘NSS-C-PE’（赤線）は C-PE におけるキューブ回転で奥行方向に $90°, 270°$ の回転のみ行わない（すなわち，奥行方向には $0°, 180°$ の回転のみ行う）変種での結果を示している．どの映像シーケンスに対しても，キューブ回転で時間断面をフレーム内に出現させる場合，時間断面ブロックが圧縮効率に影響を及ぼしているものの，回転角を制限して時間断面ブロックを出現させなければ圧縮効率は維持されることがわかる．

ここまでは，映像シーケンス全体を大きな直方体に見立てて統一的な大きさのキューブに分割して適用する手法であったが，統一的な大きさのキューブで分割していることは攻撃者にもわかる．なおかつ，キューブの大きさが分かればそのキューブ同士をマッチングするキューブベースジグソーパズル解読攻撃の根本的危険性を拭い去ることはできない．そこで，次節のピクセルキューボイドベース暗号化に拡張することで，このキューブベースジグソーパズル解読攻撃への危険性も撤廃する．

3.4　拡張版

ピクセルサブキューボイドベース暗号化 [36] のフローを図 3.7 に示す．図 3.7 から分かる通り，この暗号化はピクセルキューブベース暗号化で全体的に暗号化された映像のランダムな一部に追加で適用するものとなっている．ピクセルサブキューボイドベース暗号化の追加された全体としての拡張版を，本研究ではピクセルキューボイドベース暗号化と呼んでいる．このピクセルサブキューボイドベース暗号

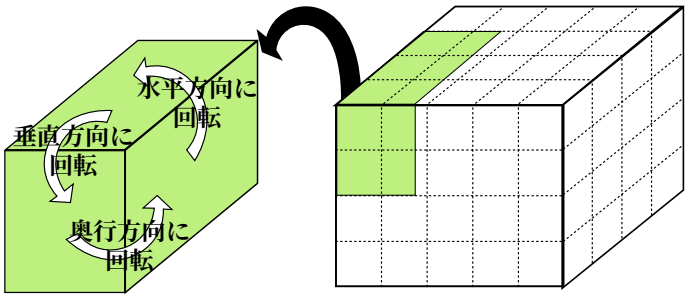


図 3.8　キューボイド回転.

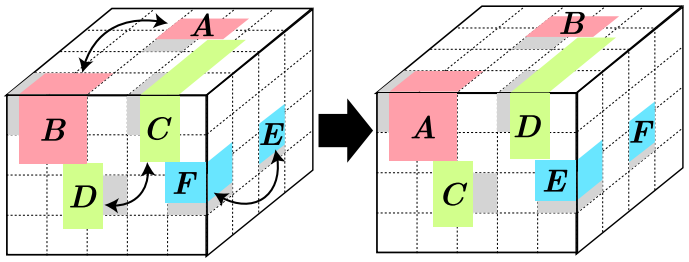
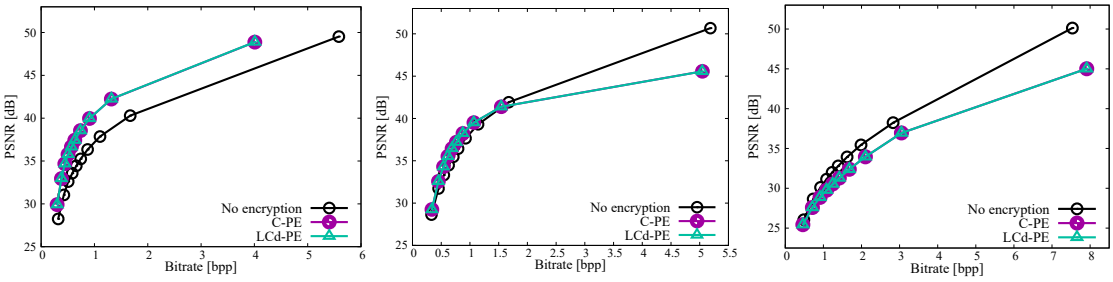


図 3.9　キューボイド入替.



(a)　(b)　(c)

図 3.10　レート歪み曲線を用いたピクセルキューブ／キューボイドベース暗号化での暗号化映像フレームの MJPEG 圧縮効率の比較：(a)*Akiyo*，(b)*Bowing*，(c)*Coastguard*.

化も，ピクセルキューブベース暗号化と同様に 4 モジュールの構成となっているが，映像全体を奥行方向には任意幅の直方体（サブキューボイド）で分割するという点で異なっている．キューボイド回転を図 3.8 に，キューボイド入替を図 3.9 に示す．キューボイド回転では，サブキューボイドを回転させる際に奥行方向には $90°, 270°$ の回転を行えないため，奥行方向には $0°, 180°$ の回転のみを行う．全体的に，ピクセルサブキューボイドベース暗号化ではキューボイド回転が MJPEG 圧縮効率を阻害しないため，全モジュール適用後も MJPEG 圧縮効率は維持される（図 3.10）．図 3.10 で，‘LCd-PE’ はピクセルキューボイドベース暗号化による圧縮効率を表す．この理由として，MJPEG では AVC，HEVC，VVC のようなフレーム間でのインター予測を行わないため，奥行方向に任意の辺幅で指定されたサブキューボイド（つまり，$16i \times 16j \times k$）で暗号化しても符号化効率にほとんど影響を及ぼさないことが挙げられる．すなわち，時間方向への辺幅が可変のサブキューボイドで暗号化できるため，単一の大きさのキューブを用いたジグソーパズル解読攻撃が意味を為さないことがわかる（図 3.11）.

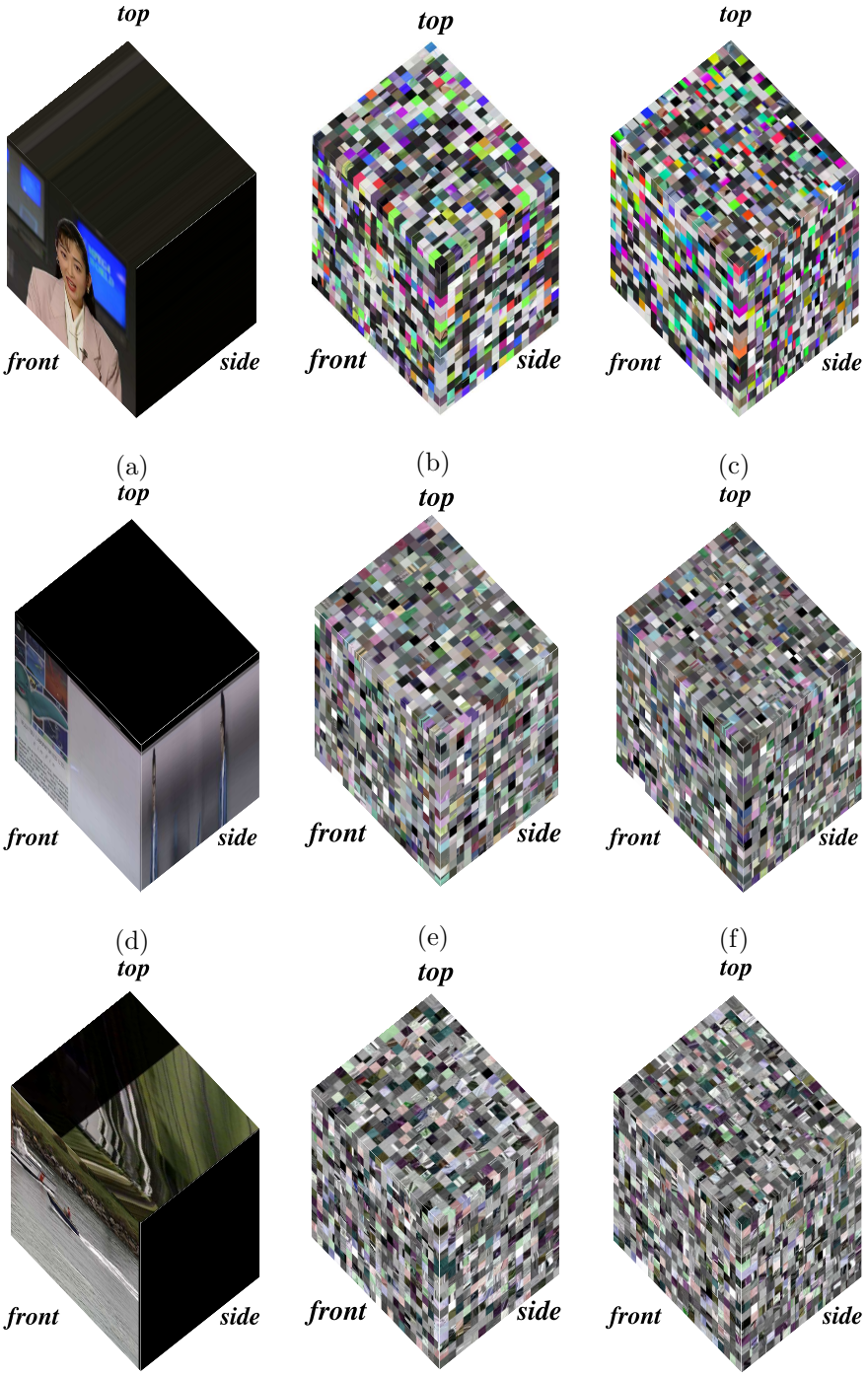


図 3.11　暗号化映像シーケンス全体の観測結果：(上行から下行に) *Akiyo*，*Bowing*，*Coastguard*，(左列から右列に) 元映像，ピクセルキューブベース暗号化による結果，ピクセルキューボイドベース暗号化による結果.

3.5　本章のまとめ

本章では，従来の MJPEG 符号化に着目したピクセルブロックベース暗号化のセキュリティを向上させるため，ピクセルキューブベース暗号化とその拡張版であるピクセルキューボイドベース暗号化を

紹介した．ピクセルキューブベース暗号化は非圧縮映像のフレーム群を一つのキューボイドに見立ててそれよりも小さなキューブに分割し，キューブ回転・キューブ入替・キューブネガポジ反転・キューブ色成分入替の四モジュールからなる暗号化を施す．キューブ回転では従来のピクセルブロックベース暗号化の持つブロック回転反転で元フレーム内に出現させられない，映像の時間方向の断面にあるブロックを元フレーム内ブロックと置き換えるため，単一フレームのみを用いた総当たり攻撃・ジグソーパズル解読攻撃といった攻撃を防止できる．キューブ回転の奥行方向への回転角が $90°, 270°$ のいずれかであった場合，時空間断面のブロックが元フレーム内に出現し，攻撃耐性向上の代わりに MJPEG 符号化効率に影響を及ぼすものの，この回転角を制限することで符号化効率への影響も効率的に抑制できることがわかった．このピクセルキューブベース暗号化が単一フレーム内でのブロックマッチングによるジグソーパズル解読攻撃を防止しても，キューブベースのジグソーパズル解読攻撃が新たに設計されると予想できるため，キューブ分割のキューボイドへの拡張であるピクセルキューボイドベース暗号化を提案した．ピクセルキューボイドベース暗号化では，ピクセルキューブベース暗号化で全体的に暗号化された映像シーケンスのランダムな一部に，キューボイド回転・キューボイド入替・キューボイドネガポジ反転・キューボイド色成分入替からなるピクセルサブキューボイドベース暗号化を適用することで，キューブベースのジグソーパズル解読攻撃さえも防止可能にした．

第 4 章

ビットキューボイドベース暗号化

本章では，画像符号化規格 JPEG のためのビットキューボイドベース暗号化 [61, 62] を提案する．ビットキューボイドベース暗号化は，(i) ビットキューボイドベース暗号化（BE: bitcuboid-based encryption）と (ii) 例外不要型符号付二進化（ESB: exception-free signed binarization） という二つの新たな考え方からなる．ビットキューボイドベース暗号化は従来の IBS [42] に動機付けられた暗号化手法であり，QDCT 係数ブロックを二進化した直方体（キューボイド）形状のビット集合，すなわち‘ビットキューボイド’を用いる．この提案法は二進領域内でのビットプレーン内・間での暗号化であり，その暗号化するための領域はビットキューボイドを細分化して得られたより小さな部分集合（‘ビットキューブ’）になる．ESB は，例外処理を用いずに二進列を十進数へ完全可逆に再十進化し，暗号化による JPEG 圧縮画像のビットレートも抑制する．JPEG 符号化内での暗号化実験では，提案法の ESB を用いた BE（BEESB: BE with BEESB）が，ビットキューブの大きさと暗号化する際の確率パラメータを多彩に用いることで柔軟な調整可能性を実現し，従来よりも効率的なビットレート増加抑制と高い攻撃耐性を有することを示す．そして，本手法を映像符号化規格 HEVC にも適用することでその汎用性を示す．

4.1 動機付け

過去十年間では，JPEG フォーマット準拠暗号化は圧縮の前後で設計することで，その圧縮効率を維持できることが多く報告されてきた [27, 35–41]．JPEG 符号化前での暗号化 [27, 35, 36] では，入替・回転反転・ネガポジ反転・色成分入替という 4 モジュールをフルカラーの画像・映像フレームに施し，別の手法 [37] では入替・回転反転・ネガポジ反転の 3 モジュールをグレースケール画像に適用することで，高い視認秘匿性と圧縮効率への親和性を実現する．JPEG 符号化後での暗号化 [38–41] では XOR やランダム入替などを JFIF 構文要素に施すことで，輝度／色差別の知覚劣化や直流／交流別の知覚劣化など，異なる種類の知覚劣化度合を実現する．

表 4.1　従来法と提案法の特性のまとめ.

手法	調整可能性	ビットレート増加	攻撃耐性
JPEG 符号化前手法 [27, 35–37]	無	**小・中量**	**高度**
JPEG 符号化後手法 [38–41]	無	**小量**	低／高度
RANDZZ (JPEG 符号化内) [22]	無	大量	**高度**
IBS (JPEG 符号化内) [42]	潜在的に適度	大量	**高度**
FIBS (JPEG 符号化内) [43]	潜在的に適度	**小量**	**高度**
DCACS (JPEG 符号化内) [44]	適度	**小量**	低度
RSF (JPEG 符号化内) [24]	僅少	**小量**	低度
提案法 BEESB (JPEG 符号化内)	**柔軟**	**小・中量**	**高度**

上記よりも古い従来法 [22–24, 42–44] は JPEG 符号化の内部で実装されていた．2.5 節で紹介したように，RANDZZ [22] は，置換攻撃に頑健であるという特長を持つ一方，JPEG ファイルのビットレートを甚大に増加させてしまうという欠点を持つ．また，RSF は JPEG 圧縮画像の圧縮効率を維持しつつ手頃に模様情報を秘匿化する一方で，置換攻撃には脆弱である．これらの手法は暗号化による知覚劣化度合の調整可能性をほとんど／全く持たないが，他の数手法 [42–44] は潜在的に幾分かの調整可能性を持つ．IBS [42] は，多少の調整可能性を持ちつつ置換攻撃にも頑健である一方で，JPEG 圧縮画像の甚大なビットレート増加を抑えられない．FIBS [43] と DCACS [44] でも多少の調整可能性を潜在的に実現可能であり，JPEG 圧縮画像のビットレート増加も抑制できる一方，FIBS でさえ一部の係数しか入れ替えない場合非暗号化部分の漏洩に脆弱であるなど，結局どちらも置換攻撃に脆弱となる．また，JPEG 符号化内フォーマット準拠暗号化は，FIBS で問題視されていたスケッチ攻撃への耐性を実は既に有している．その理由はこれらの手法では各ブロック内の詳細なテクスチャ情報を保護できるためであり，そのような手法に対してブロック間での漏洩に気を配ることは的外れといえる．そして，保護されたブロック同士をマッチングしようとするジグソーパズル解読攻撃も意味をなさないことは明白であり，それらはジグソーパズル解読攻撃に対しても頑健といえる．このように JPEG 符号化内フォーマット準拠暗号化は JPEG 符号化効率を阻害するうえに種々の攻撃に脆弱性を持つことから，昨今あまり関心を持たれていないものの，QDCT 係数領域で注意深く選定された部分のビットを暗号化することで，攻撃耐性と圧縮効率の維持を両立しつつ，知覚劣化度合に柔軟な調整可能性を持たせられる．

これより，本研究では柔軟に調整可能な JPEG 符号化内フォーマット準拠暗号化を提案することで，攻撃耐性と圧縮効率の維持を両立しつつ，従来よりも多くのセキュリティ的要求に応えることに貢献する．提案法と従来法の特性の比較を表 4.1 にまとめる．

表 4.2　$D = 10$ ビットの場合における，各十進数に対応した FSB と ESB による二値列（符号ビット $|D-1$ 絶対値ビット列）．

十進数	FSB 二進化による二値列	ESB 二進化による二値列
511	0|111111111	0|111111111
510	0|111111110	0|111111110
509	0|111111101	0|111111101
⋮	⋮	⋮
3	0|000000011	0|000000011
2	0|000000010	0|000000010
1	0|000000001	0|000000001
0	0|000000000	0|000000000
−1	1|000000001	1|000000000†
−2	1|000000010	1|000000001
−3	1|000000011	1|000000010
⋮	⋮	⋮
−510	1|111111110	1|111111101
−511	1|111111111	1|111111110
−512	1|000000000*	1|111111111

* -2^{D-1} に再十進化される二値列
† -1 に再十進化される二値列

4.2　ビットキューボイドベース暗号化

JPEG のエントロピー符号化では，QDCT 係数領域が疎になるほど効率的な圧縮を行えるものの，ビット単位での暗号化がこれを密にすればするほど圧縮に不都合となり，とくに何の考慮もせずに暗号化すると，JPEG の符号化効率を大きく妨げることになる．なかでも，一つの係数が m ビットの非ゼロビットを持ち，その係数中のビットが他の m 個のゼロ係数との間で入れ替えられたとき，これら $m+1$ 個中の m 個の係数が非ゼロとなりうる．すなわち，

$$\mathbf{b}_0 = b_{n-1} \cdots b_0 \quad \text{where} \sum_{i=0}^{n-1} b_i = m \tag{4.1}$$

という二値列を持つ係数が $m+1$ 個の係数の二値列 $\{\mathbf{b}_t\}_{t=0}^{m}$ $(\mathbf{b}_1, \cdots, \mathbf{b}_m = \mathbf{0})$ との間でビット単位に入れ替えられた場合，いずれかの m 個の係数は二値列に非ゼロビットを含み，再十進化で非ゼロ係数になる可能性がある．非ゼロ係数の個数増加は圧縮効率を阻害することに起因する．ビット単位での細かなシャッフルは暗号化デコード画像の知覚劣化における柔軟な調整可能性を達成できるものの，圧縮

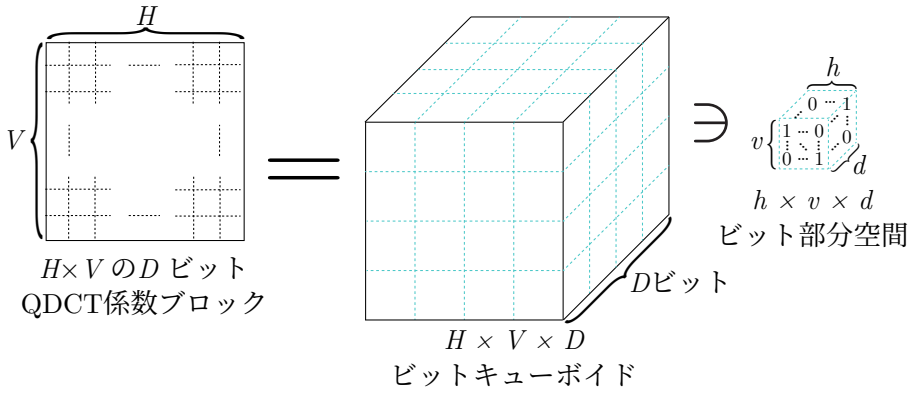


図 4.1　ビットキューボイドとその部分（空間内の）集合.

効率維持のためには非ゼロ係数の個数は暗号化前後で大きく変わるべきではないということになる．また，暗号化されていない非ゼロビットは元の画像特徴を含んでいると考えられ，そのようなビットが残っていると，暗号化部分の削除によって元の画像特徴が漏洩する危険性を孕んでいる．これらのことから本研究では，攻撃耐性と圧縮効率を両立するために，二進空間内での小さな部分空間内で非ゼロビットをシャッフルする．

提案法のビットキューボイドベース暗号化とは，$H \times V$ 個の D ビット十進数を $H \times V \times D$ の直方体形状のビット集合（‘ビットキューボイド’）に変換し，その中で各 $h \times v \times d$ $(\geq 2;\ h \leq H, v \leq V, d \leq D)$ ビット部分集合内（図 4.1）の位置関係をランダムに変更する手法である．ビットキューボイドとその部分集合は任意サイズに設定できるものの，本研究では簡単さと JPEG 符号化効率のために $H = V = 8, D = 10, h = v = d$ とし，この $h = v = d$ である部分集合を ‘ビットキューブ’ と呼ぶ．さらに，m 個の非ゼロビットを持つ h^3 ビットキューブ

$$\mathbf{B}_m := b_{h^3-1} \cdots b_0 \text{ where } \sum_{i=0}^{h^3-1} b_i = m \tag{4.2}$$

を ‘m-キューブ’ と称する．暗号化後の m-キューブ $\mathbf{B}'_m$ は

$$\mathbf{B}'_m = \mathbf{B}_m \mathscr{P}_{h^3} \tag{4.3}$$

で得られる．ここで，$\mathscr{P}_{h^3}$ は $h^3 \times h^3$ のランダム置換行列であり，PRNG [50] からの擬似乱数列を用いて計算される．

ビット単位での位置関係を変更する際，ビットキューブ内の総ビット数を用いることができ，総ビット数を用いた暗号化によって提案法では暗号化デコード画像の知覚劣化度合に調整可能性を持たせられる．この操作は符号化効率を甚大には阻害しない．(4.2) により，暗号化後の m-キューブ $\mathbf{B}'_m$ も，元の m-キューブ $\mathbf{B}_m$ と同様に $\sum_{i=0}^{h^3-1} b_i = m$ を満たす．QDCT 係数領域は極めてスパースな領域であるため，この提案法がゼロ係数を非ゼロ係数に変えるとしても，別の非ゼロ係数はゼロ係数になりうる．つまり，符号化効率を阻害しうるゼロ係数の割合がほとんど変化しない．実際の有効性は 4.3.2 節で議論

する．

4.2.1 例外不要型符号付二進化

固定長符号付二進化における問題点

JPEG QDCT 係数領域で調整可能な暗号化を設計する場合，符号付十進数である QDCT 係数を二進領域に変換しその二進領域内で適切な粒度によって暗号化することで，知覚劣化度合に調整可能性を持たせられる．ただし，二値列が何らかの可変長符号化の前で暗号化され，かつその符号化が十進数を入力とする（たとえば JPEG でのハフマン符号化などが適用される）場合，暗号化二値列は符号化のために一旦再十進化されることが望まれる．しかし，暗号化二値列が 1 ビットの負号ビットと $D-1$ ビットの ‘全てゼロの’ 絶対値ビット列から構成される，すなわち

$$\underbrace{1}_{\text{負号ビット}} \;\Big|\; \underbrace{0\cdots0}_{\substack{D-1\\ \text{絶対値ビット列}}} \tag{4.4}$$

という列になる場合，この二値列が通常の二進-十進変換の

$$c = (-1)^{b_{n-1}} \left(\sum_{i=0}^{n-2} b_i \times 2^i \right) \tag{4.5}$$

を用いて再十進化された際に $-1 \times 0 = 0$ となり（つまり，負号ビットが消失し），暗号化係数が平文化できなくなる．この非常的な二値列を再十進化するための例外処理として，(4.4) の二値列を -2^{D-1} と定義する．デコーダ側の平文化器では，例外的に再十進化されたはずの -2^{D-1} は何の補助情報も要さずに，一度の符号ビット抽出と $D-1$ 回の絶対値ビット列抽出によって

$$\underbrace{1}_{\text{負号ビット}} \;\Big|\; \underbrace{\not{1}}_{\substack{D\text{ 番目絶対値}\\ \text{ビット（破棄）}}} \underbrace{0\cdots0}_{\substack{D-1\text{ 絶対値}\\ \text{ビット列}}} \;\Rightarrow\; 1|0\cdots0 \tag{4.6}$$

と再び二進化される．D 番目絶対値ビットでようやく現れる 1 は，$D-1$ 回の絶対値ビット列抽出まで必ず抽出できない（全て 0 が取り出される）ため，暗号化器側と同一の二値列が得られ完璧に平文化される．

ただし，上記の例外処理は JPEG 符号化に対し悪い効果を持つ．再十進化値の -2^{D-1} の絶対値ビット長には JPEG のハフマン符号表（Annex C）で最長の符号語が割り当てられるため，この最長の符号語に -2^{D-1} の可変長ビット列が連結・格納されることにより，暗号符号化ビットストリームのビットレートを増加させてしまう．この増加は $1|0\cdots0$ の二値列が出現するたびに引き起こされ，そのような二値列はスパースな（絶対値ビット列に 0 の多い係数が遍在する）JPEG QDCT 係数領域では頻出するため，結果的に JPEG ファイルをサイズ増加させてしまうことになる．たとえば，二組の十進数 $(-1, 0)$ から二進化された二値列 $(1|000000001, 0|000000000)$ の最下位ビット（LSB）同士が $(1|000000000, 0|000000001)$ と入れ替えられた場合，この組は $(-512, 1)$ に再十進化される（表 4.2 二列目）．

提案の二進化

BE では，実際の暗号化／平文化の前後で二進化・再十進化を伴う．その理由は以下である：

- ビット単位で QDCT 係数を暗号化するために，その処理の前後で係数を二進化する必要がある．
- QDCT 係数を後段の処理へ容易に入力するために，暗号化後の二値列を再十進化する必要がある．

ただし，4.2.1 節でも述べたように，単純なビットキューボイドベース暗号化に FSB を用いた場合 [61] では再十進化に例外処理を要してしまう．この問題を解決するために，本研究では ESB を提案する．

ESB とは，十進数 c を n ビットの二値列 $\mathbf{b}$ に変換する FSB であり，

$$\mathbf{b} = \mathrm{sgn}(c) | \left(\mathrm{abs}(c) - \mathrm{sgn}(c)\right)_{2,n-1} \tag{4.7}$$

の形式で二進化を行う．(4.7) で，符号ビット $\mathrm{sgn}(c)$ を減算することは負の二値列を全体的に一つずつシフトすることに相当する．$\mathbf{b}$ は c に，

$$c = (-1)^{b_{n-1}} \left(\left(\sum_{i=0}^{n-2} b_i \times 2^i \right) + b_{n-1} \right) \tag{4.8}$$

と再十進化される．(4.8) で，二進化後の符号 $b_{n-1} = \mathrm{sgn}(c)$ を再十進化後の絶対値部分 $\sum_{i=0}^{n-2} b_i \times 2^i$ に加算することによって，$\mathbf{b}$ は c に例外処理を要さず再十進化される．

ESB では，二値列 $1|0\cdots0$ を負数 -1 に再十進化する（表 4.2）ので，この再十進化値は最短ビット長の可変長ビット列として JFIF ファイル構造 [63] の ECS セグメント内に格納される．そのため，ESB では FSB で引き起こされる JPEG の符号化効率阻害を防止できる．[*1]

本研究ではこの BEESB を JPEG の QDCT 係数領域で適用することを考え，暗号化の前には 19936 ビットの状態空間を持つ MT を 256 ビットの SHA-2 ハッシュ値（シード）で初期化する（図 4.2）．QDCT 係数領域の各 8×8 ブロックに存在する十進数を ESB で二進化し，暗号化対象の m-キューブを検出して肝心の暗号化／平文化を MT 乱数列を用いて行い，再十進化を行って暗号化後の十進数ブロックに変換する．暗号化と平文化で用いるランダム置換は，それを生成するための乱数列が同一のキーから得られるために同一となる．

4.2.2 柔軟な調整可能性

BEESB は特定のビットキューブを暗号化することで柔軟な調整可能性を実現でき，その実現は以下の考え方による：

- 任意の m-キューブ（$m \leq h^3 - 1$）を組み合わせつつ暗号化する．

[*1] ESB はシンプルな二進化アルゴリズムであるため，従来法 RANDZZ [22], IBS [42], DCACS [44] などにも容易に応用できる．

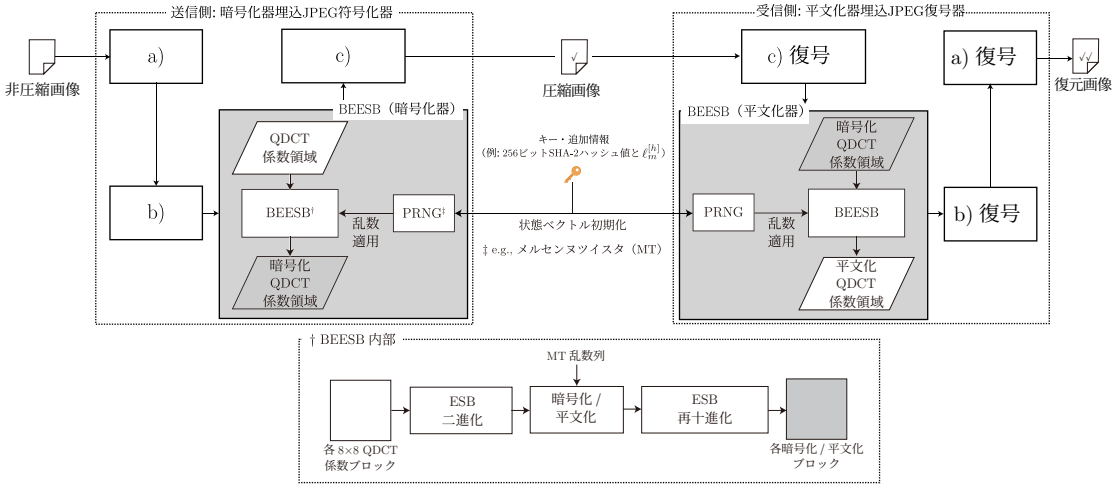


図 4.2　JPEG 符復号器に埋め込まれたビットキューボイドベース暗号化のフロー.

(a)　(b)　(c)　(d)

図 4.3　サイズ 2^3 における，多種の m-キューブを用いた場合での，BEESB の引き起こす知覚劣化（JPEG $Q = 70$）：(a) 未暗号化状態の *ucid00059* の一部, (b) $\ell_1^{[2]} = 100$ での暗号化結果, (c) $\ell_2^{[2]} = 100$ での暗号化結果, (d) $\ell_3^{[2]} = 100$ での暗号化結果.

(a)　(b)

図 4.4　FSB と ESB を用いた場合における，全 m-キューブ（サイズ 2^3）の暗号化結果の違い（JPEG $Q = 70$ and $\ell_{1,\cdots,7}^{[2]} = 100$）：(a) FSB と (b) ESB.

- 特定の m-キューブを確率（$\ell_m (\in \mathbb{Z}_{[0\ 100]})$ %）で暗号化する.

上記の二調整で l 種の m-キューブを暗号化する場合,

$$\mathcal{L} = \prod_{i=1}^{l} \max \ell_{m_i} = 100^l \quad (m_i \in \mathbb{Z}_{[0\ h^3]},\ l \leq h^3 - 1) \tag{4.9}$$

通りの調整が実現できる．たとえ少ない個数の m-キューブを暗号化したとしても，その中でその m-キューブは調整可能な知覚劣化度合を実現する．[*2]

4.2.3 セキュリティ

前述した通り，フォーマット準拠暗号化には置換攻撃や総当たり攻撃・スケッチ攻撃・ジグソーパズル解読攻撃などが報告されており，いずれも暗号化コンテンツを元コンテンツに平文化キー無しで直接復元／近似する ‘暗号文単独攻撃’ に分類されている．本節ではこれらの攻撃に対する提案法の耐性を理論的に考察し，実際の耐性と統計的性質は 4.3.3 節で実験的に示す．

置換攻撃に対する頑健性

まず，置換攻撃に対する耐性を考察する．BEESB では元画像の特徴を多く含んでいる m-キューブを暗号化し，置換攻撃ではその暗号化された m-キューブを削除する．その結果，削除された m-キューブは元画像の特徴を含んでいた m-キューブとなり，BEESB は置換攻撃から元画像の特徴を保護できることになる．換言すれば，削除対象の m-キューブを暗号化することで，削除された後には元画像の特徴をそもそもほとんど含まないように設計できるのである．低圧縮（高品質）になるほど，粗く量子化されないことによって暗号化対象の m-キューブが増加し，BEESB ではより多くの ‘特徴的な m-キューブ’ を暗号化できるようになる．

総当たり攻撃に対する頑健性

続いて，総当たり攻撃に対する耐性を考察する．暗号化対象の m-キューブの個数が増加するほど，BEESB ではこの攻撃に強い耐性を持つ．h^3 ビットのつまった各ビットキューブ内の正しい配置を特定するための総当たりは 2^{3h} 回であることから，N 個の暗号化 m-キューブに対しては $\left(2^{3h}\right)^N = 2^{3hN}$ 回の総当たりが要される．また，本研究では安全性の基準となる SHA-256 ハッシュ値 [52] をキーに用いるため，このキーに対しては 2^{256} 回の総当たりを要することから，$2^{3hN} \geq 2^{256}$，すなわち $3hN \geq 256$ である場合に頑健性が保証される．加えて，暗号化前後の二値列は区別が困難であることから，以下のヒントのみから正しい QDCT 係数の集合を見つけることは極めて困難といえる：

- 各ブロック内で DC 係数が最大の絶対値を持つ．
- オフセット処理（レベルシフト）なしで DC 係数が負値になることは通常ない．

これらの情報が，もとの QDCT 係数の集合をさらに発見困難にしている．したがって，攻撃者は尤もらしい QDCT 係数の集合を見つけることすら困難といえるものの，暗号化後の QDCT 係数が異なる

[*2] 自明だが，上記のほかにも BEESB で調整可能性を実現する方法は無数にある．たとえば，ビットキューボイドごとに ℓ_m の選択を変えることや，ビットキューブを隣り合うビットではなく異なる位置からのビットでランダムに構成すること，ビットキューブの暗号化を（シャッフル以外に）変えること，などが挙げられる．

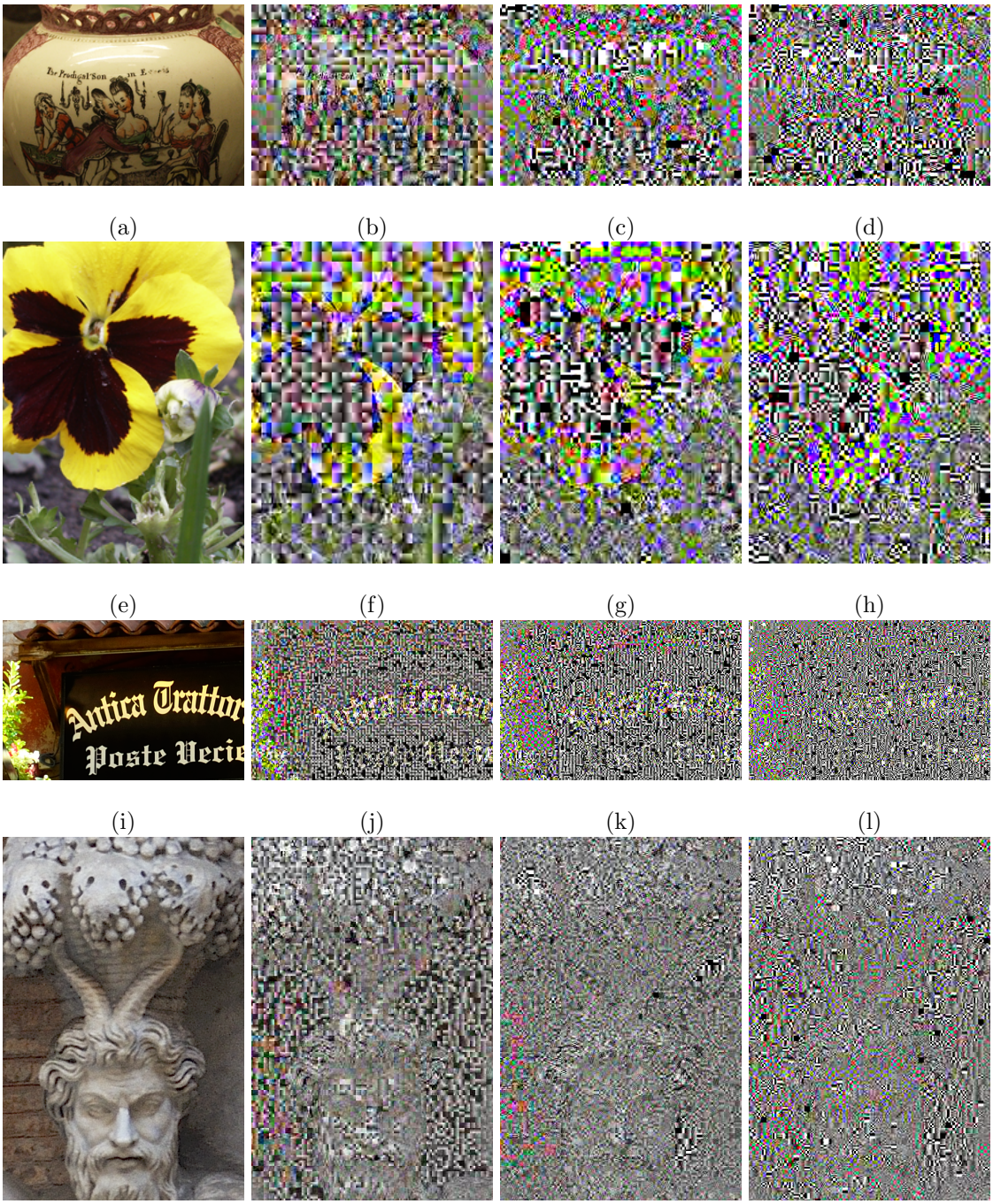
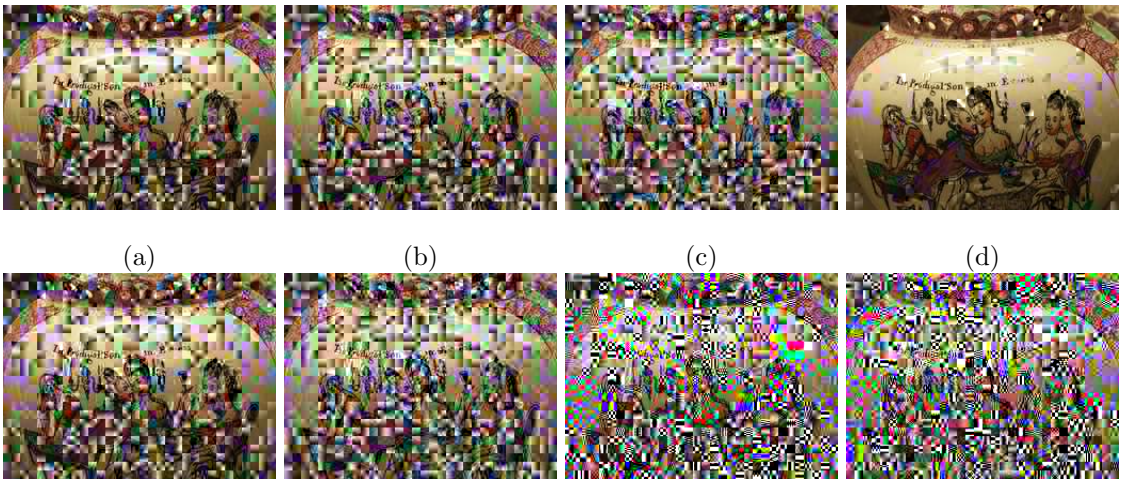


図 4.5　異なる大きさの m-キューブを用いた BEESB による知覚劣化度合の違い (JPEG $Q = 70$) : (a-d) *ucid00059* の一部, (e-h) *ucid00069* の一部, (i-l) *r00444b95t* の一部, (m-p) *r00b8d4a2t* の一部, (a,e,i,m) 元画像, (b,f,j,n) $\ell^{[2]}_{1,\cdots,7} = 100$ での暗号化結果, (c,g,k,o) $\ell^{[3]}_{1,\cdots,26} = 100$ での暗号化結果, (d,h,l,p) $\ell^{[4]}_{1,\cdots,63} = 100$ での暗号化結果.

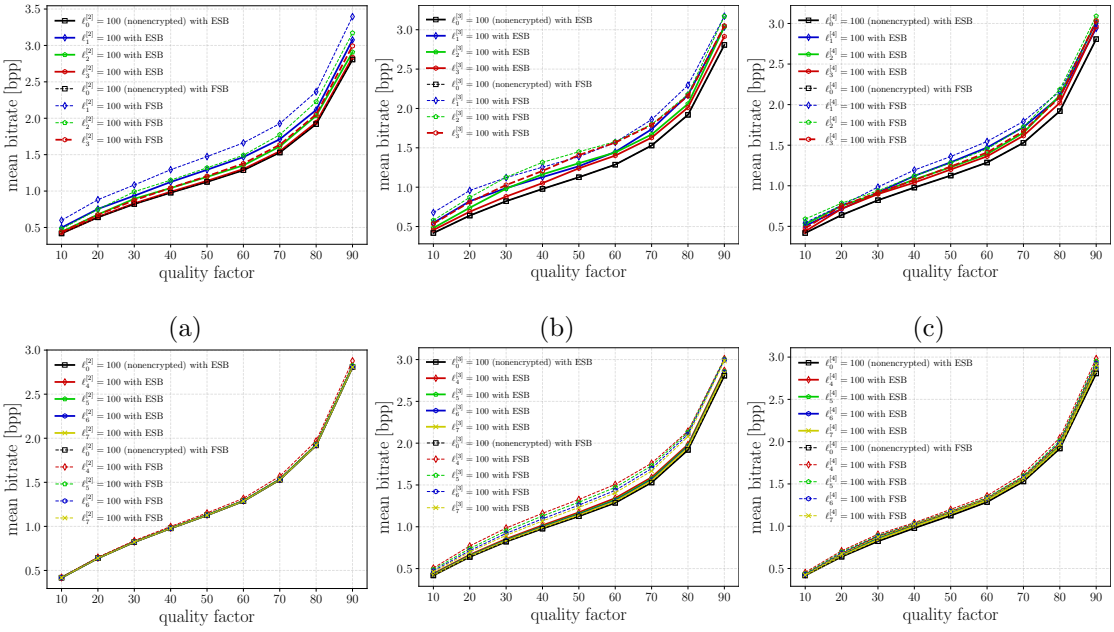
キーでも偶然復元されてしまう可能性は否定しがたい．このため，本研究では異なるキーが用いられた際の真正のキーとの過敏性も検証する．この検証は 4.3.3 節で議論する．



(a) (b) (c) (d)

(e) (f) (g) (h)

図 4.6　異なる大きさ・確率で m-キューブを暗号化する BEESB による知覚劣化度合の違い (JPEG $Q = 70$)：(a) $\ell_1^{[2]} = 50$ での暗号化結果, (b) $\ell_1^{[2]} = 70$ での暗号化結果, (c) $\ell_1^{[2]} = 90$ での暗号化結果, (d) $\ell_2^{[2]} = 50$ での暗号化結果, (e) $\ell_{1,2,3}^{[2]} = 50$ での暗号化結果, (f) $\ell_1^{[2]} = 70, \ell_2^{[2]} = 90, \ell_3^{[2]} = 100$ での暗号化結果, (g) $\ell_{1,2,3}^{[2]} = 50$ & $\ell_{1,2,3}^{[3]} = 50$ & $\ell_{1,2,3}^{[4]} = 50$ での暗号化結果, (h) $\ell_{1,2,3}^{[2]} = 80$ & $\ell_{1,2,3}^{[3]} = 60$ & $\ell_{1,2,3}^{[4]} = 40$ での暗号化結果.



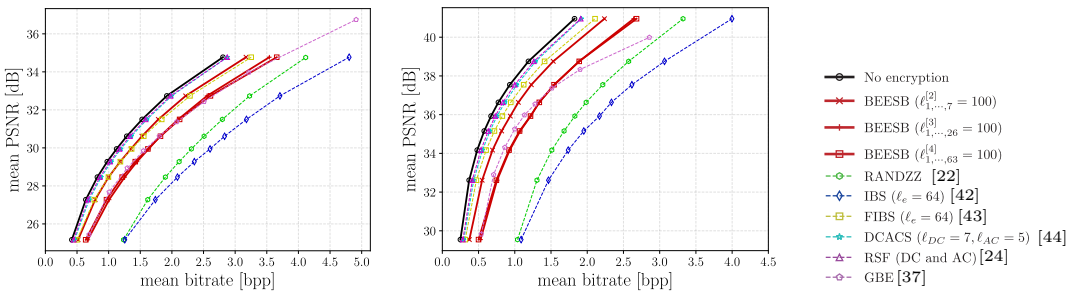
(a) (b) (c)

(d) (e) (f)

図 4.7　BEESB/BEFSB: BE with FSB による暗号化を埋め込んだ場合での JPEG 符号化効率の比較: (a) $\ell_0^{[2]}, \cdots,$ or $\ell_3^{[2]} = 100$ による結果, (b) $\ell_0^{[3]}, \cdots,$ or $\ell_3^{[3]} = 100$ による結果, (c) $\ell_0^{[4]}, \cdots,$ or $\ell_3^{[4]} = 100$ による結果, (d) $\ell_4^{[2]}, \cdots,$ or $\ell_7^{[2]} = 100$ による結果, (e) $\ell_4^{[3]}, \cdots,$ or $\ell_7^{[3]} = 100$ による結果, (f) $\ell_4^{[4]}, \cdots,$ or $\ell_7^{[4]} = 100$ による結果.

スケッチ攻撃・ジグソーパズル解読攻撃に対する頑健性

さいごに，スケッチ攻撃・ジグソーパズル解読攻撃への耐性も考察する．BEESB ではブロックごとの詳細な模様情報を暗号化することを断っておく．スケッチ攻撃では，8×8 の各 QDCT 係数ブロッ



(a) (b)

図 4.8　提案法・従来法を用いた場合における R-D 曲線を用いた JPEG 符号化効率の比較: (a) UCID データセット・(b) RAISE データセット.

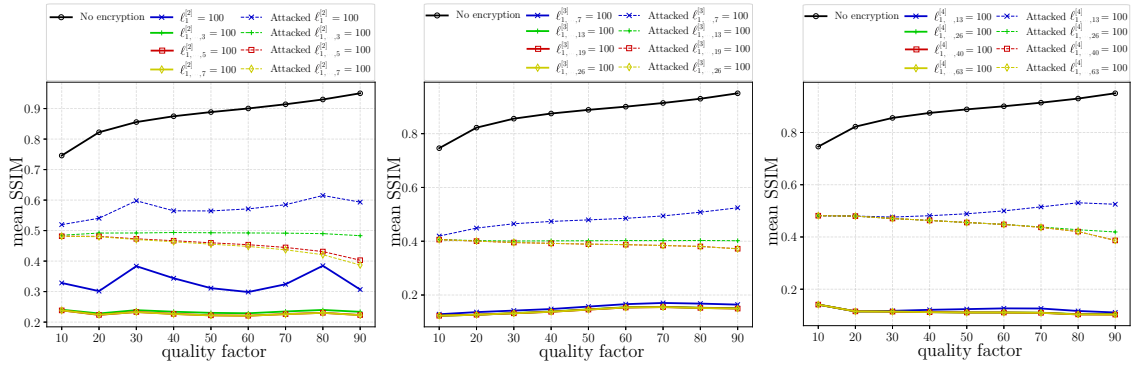
クが元と同じ位置にある（シャッフルされていない）場合，その隣接（アウトライン）情報を漏洩しようとする．しかし，漏洩されるアウトラインはブロック同士が繋ぎ合わさったときの大域的な情報であり，BEESB で暗号化後のブロック内に含まれる局所的な（詳細な）模様情報は未だ保護されている．このアウトライン情報さえも保護するためには，ブロック間でのシャッフル [39] が最も単純かつ率直なアプローチとして既出である．また，暗号化後のブロックをジグソーパズル解読攻撃で正しく連結させられないことも明らかである．これらのことから，本研究ではスケッチ攻撃・ジグソーパズル解読攻撃に関してこれ以上深掘りしない．

一方，暗号化デコード画像のヒストグラムから，攻撃者が元画像に関する情報を推測する可能性のあることから，本研究では実験的にヒストグラム解析も行う．ヒストグラム解析についての言及は 4.3.3 節に示される．

4.3　実験

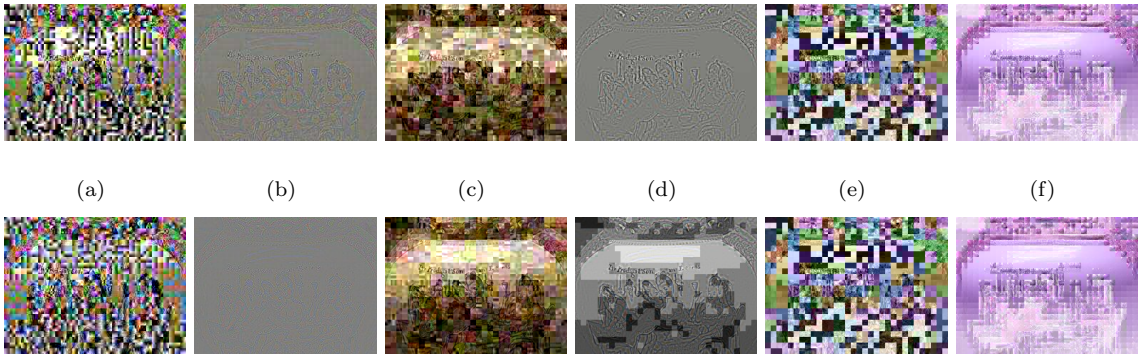
提案法の有効性を示すため，本研究では大別して三つの実験を行った．4.3.1 節では，異なるサイズ・種類の m-キューブを用いた場合での，様々な確率パラメータ $\ell_m^{[h]}$ を用いた知覚劣化度合の調整可能性を示す．ここで，$\ell_m^{[h]}$ とはサイズ h^3 で分割された m-キューブを何 % の確率で暗号化するかという確率パラメータである．4.3.2 節では，従来法 [22, 37, 42–44] と比較して提案法での暗号化 JPEG 圧縮画像の圧縮効率を調査する．4.3.3 節では，暗号化部分を削除したり誤ったキーを用いたりすることで置換攻撃・総当たり攻撃に対する提案法の頑健性を調査し，提案法による暗号化デコード画像のヒストグラムを解析した．実験は UCID データセット [64] から 100 枚と RAISE データセット [65] から 20 枚の入力画像に基づき，JPEG の参照ソフトウェア *libjpeg* [66] を用いて行った．以降特筆しない限り，実験は UCID データセットからの画像に基づくものとする．提案法 BEESB と従来の JPEG 符号化内フォーマット準拠暗号化 [22, 42–44] に関して，実験の手順は以下となる：

1. 入力画像を JPEG 符号化し，符号化中の QDCT 係数を暗号化する．JPEG の品質ファクタは



(a) (b) (c)

図 4.9　様々なサイズ・種類の m-キューブを暗号化した際の BEESB における知覚劣化度合・置換攻撃による復元品質: (a) 2^3 ビットキューブでの結果, (b) 3^3 ビットキューブでの結果, (c) 4^3 ビットキューブでの結果.



(a) (b) (c) (d) (e) (f)

(g) (h) (i) (j) (k) (l)

図 4.10　BEESB ($\ell^{[2]}_{1,\cdots,7} = 100$)・DCACS ($\ell_{DC} = 7, \ell_{AC} = 5$)・RSF (DC and AC) で暗号化された画像と，置換攻撃による復元画像: (a) BEESB ($Q = 50$), (b) 攻撃後の (a), (c) DCACS ($Q = 50$), (d) 攻撃後の (c), (e) RSF ($Q = 50$), (f) 攻撃後の (e), (g) BEESB ($Q = 90$), (h) 攻撃後の (g), (i) DCACS ($Q = 90$), (j) 攻撃後の (i), (k) RSF ($Q = 90$), (l) 攻撃後の (k).

$Q = 10, 20, \cdots, 90$ とする.

2. 1. で得られた JPEG 圧縮画像・暗号化 JPEG 圧縮画像のビットレートを計測する.
3. これらを JPEG 復号し，復号内で暗号化 QDCT 係数を平文化する.
4. 3. での平文化デコード画像・暗号化デコード画像の元画像に対する PSNR・SSIM [67] を計算する.
5. 1. から 4. までを全入力画像に適用することで，平均の結果を算出する.

また，最新の従来法 [37] は上記と異なり，画像のピクセル領域（JPEG 符号化の外側）で適用した.

4.3.1 柔軟な調整可能性

まず，BEESB によってもたらされる知覚劣化を確認した．この実験では，8×8 の 10 ビット QDCT 係数ブロックから MSB（符号ビット）プレーンと LSB プレーンを除いて得られる $8\times8\times8$ ビットキューボイドを使用し，そのビットキューボイドを 64 個の非重複な 2^3 ビットキューブに分割した．この条件では，七種の m-キューブ（$m=1,\cdots,7$）を暗号化することができ，0-キューブ・8-キューブに関しては内部が全て 0 ないし 1 で満たされているためにシャッフルによる暗号化の効果は得られない．図 4.3 に示されるように，暗号化対象の m-キューブを変えることで異なる度合の知覚劣化が引き起こされる．$1,2,3$-キューブは比較的大きな知覚劣化を生じさせるのに対してその他の $4,\cdots,7$-キューブは無視できるほど小さな知覚劣化度合であることが実験的にわかったため，本研究では $4,\cdots,7$-キューブの暗号化結果は除いている．図 4.4 に示されるように，$1,\cdots,7$-キューブを組み合わせて暗号化することで最も強い知覚劣化が引き起こされる．BEESB での二進化に FSB を用いていた BEFSB [61] では，例外処理によって再十進化された異常値 -2^{D-1} の影響で，暗号化画像の色・模様が両方大きく変化していることがわかるが，BEESB では異常値を生じさせないためにそのような大きな変化はない．BEFSB による望ましくない劣化が引き起こされる場合に，その劣化のもととなる異常値を用いて新たな攻撃が生みだされる可能性があるが，BEESB では異常値を発生させないためにそのようなリスクがない．

次いで，サイズの異なるビットキューブを用いた BEESB による効果を調査した．図 4.5 に示されるように，ビットキューブサイズの変動によって BEESB による知覚劣化度合の強化されることがわかる．2^3 ビットキューブで分割された m-キューブの暗号化では，LSB プレーンに存在するビット同士の位置関係を変えないために，元画像の概形をわずかに残している．3^3 ビットキューブで分割された m-キューブの暗号化では，LSB プレーンのビットも暗号化するために，より強大な知覚劣化を生じさせている．4^3 ビットキューブで分割された m-キューブの暗号化では，2^3 ビットキューブでの場合のように LSB プレーンを暗号化していないものの，2^3 よりも大きな部分空間内でビット同士の位置関係を入れ替えるために，最も強い知覚劣化を引き起こしている．

さいごに，様々な設定の組合せによる効果を調査した．図 4.6 に示されるように，m-キューブの大きさを変え，分割を重複させ，異なる確率パラメータを用いることによって，BEESB では見事に柔軟な調整可能性を実現していることがわかる．2^3 ビットキューブの場合，1-キューブか 2-キューブのみの暗号化では限られた調整可能性しか施せない（図 4.6(a-d)）のに対し，$1,2,3$-キューブの確率を変えて組み合わせることによって知覚劣化度合が柔軟に調整される（図 4.6(e, f)）．さらに，暗号化対象 m-キューブの大きさを変えて重複させることによって，非常に柔軟な調整可能性が生まれる（図 4.6(g, h)）．

表 4.3　BD 値を用いた BEESB と従来法での各圧縮効率（UCID データセットでの結果）：(a) BEESB ($\ell^{[2]}_{1,\cdots,7} = 100$), (b) BEESB ($\ell^{[3]}_{1,\cdots,26} = 100$), (c) BEESB ($\ell^{[4]}_{1,\cdots,63} = 100$), (d) RANDZZ, (e) IBS ($\ell_e = 64$), (f) FIBS ($\ell_e = 64$), (g) DCACS ($\ell_{DC} = 7, \ell_{AC} = 5$), (h) RSF, (i) GBE.

手法	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
平均 BD-PSNR [dB]	−0.90	−1.96	−1.87	−4.23	−4.81	−0.93	−0.31	**−0.21**	−1.79
平均 BD-rate [%]	19.19	44.18	42.80	106.44	129.22	19.99	6.22	**4.25**	41.42

表 4.4　BD 値を用いた BEESB と従来法での各圧縮効率（RAISE データセットでの結果）：(a) BEESB ($\ell^{[2]}_{1,\cdots,7} = 100$), (b) BEESB ($\ell^{[3]}_{1,\cdots,26} = 100$), (c) BEESB ($\ell^{[4]}_{1,\cdots,63} = 100$), (d) RANDZZ, (e) IBS ($\ell_e = 64$), (f) FIBS ($\ell_e = 64$), (g) DCACS ($\ell_{DC} = 7, \ell_{AC} = 5$), (h) RSF, (i) GBE.

手法	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	(i)
平均 BD-PSNR [dB]	−1.87	−3.43	−3.39	−6.44	−7.41	−1.18	−0.69	**−0.50**	−3.02
平均 BD-rate [%]	39.04	81.78	79.98	183.90	226.62	23.07	13.03	**9.34**	79.91

4.3.2　符号化効率

まず，BEESB と BEFSB を用いた場合の暗号化 JPEG 圧縮画像のビットレート増加量を比較した．ビットキューブのサイズによらず，BEESB では BEFSB よりも効率的にビットレート増加を抑制した（図 4.7）．$1, 2, 3$-キューブでの場合，BEFSB でもビットレート増加を抑制しているものの，BEESB ではそれよりもさらに抑制することがわかり，BEFSB のおよそ半分ほどに抑えることがわかる（図 4.7(a-c)）．$4, \cdots, 7$-キューブでの場合，BEFSB では未だビットレート増加に悪影響を及ぼしているが，BEESB ではもはやその影響がないことがわかる（図 4.7(d-f)）．

また，BEESB における符号化効率を従来法と比較した．比較の際に，レート歪み曲線を描画し，Bjøntegaard delta（BD）値 [68] を計算した．本実験では RANDZZ [22], IBS [42], FIBS [43], DCACS [44], RSF [24] を従来の JPEG 符号化内フォーマット準拠暗号化として比較に用い，グレースケールブロックベース暗号化（GBE: grayscale block-based encryption） [37] を最新の従来法として比較に用いた．そして，提案の ESB を FIBS・RSF 以外の JPEG 符号化内フォーマット準拠暗号化に用いた．BEESB が RANDZZ・IBS よりも良好な符号化効率を実現することが図 4.8・表 4.3,4.4 よりわかる．ここで，表 4.5 に調整パラメータ ℓ_e, ℓ_{DC}, and ℓ_{AC} の意味を示す．とりわけ，2^3 ビットキューブでの BEESB は GBE よりも両データセットで良好な符号化効率，FIBS よりも UCID データセットで良好な符号化効率を実現することがわかる．BEESB よりも良好な符号化効率を示す DCACS・RSF については，4.3.3 節で説明の置換攻撃に耐性を持たず，本研究での目的である調整可能性に乏しいことを断っておく．また，RAISE データセットなどの高解像度画像データセットに対して FIBS が BEESB

表 4.5　従来の JPEG 符号化内フォーマット準拠暗号化に用いられる調整パラメータの意味.

調整パラメータ	意味
ℓ_e $(\in \mathbb{Z}_{[1\ 64]})$	各ブロック内で DC 係数からジグザグスキャン順に暗号化する QDCT 係数の個数.
ℓ_{DC} $(\in \mathbb{Z}_{[1\ 7]})$	LSB プレーンから暗号化する DC 係数ビットプレーンの数.
ℓ_{AC} $(\in \mathbb{Z}_{[1\ 5]})$	各ブロック内で DC 係数からジグザグスキャン順に暗号化する AC 係数の個数.

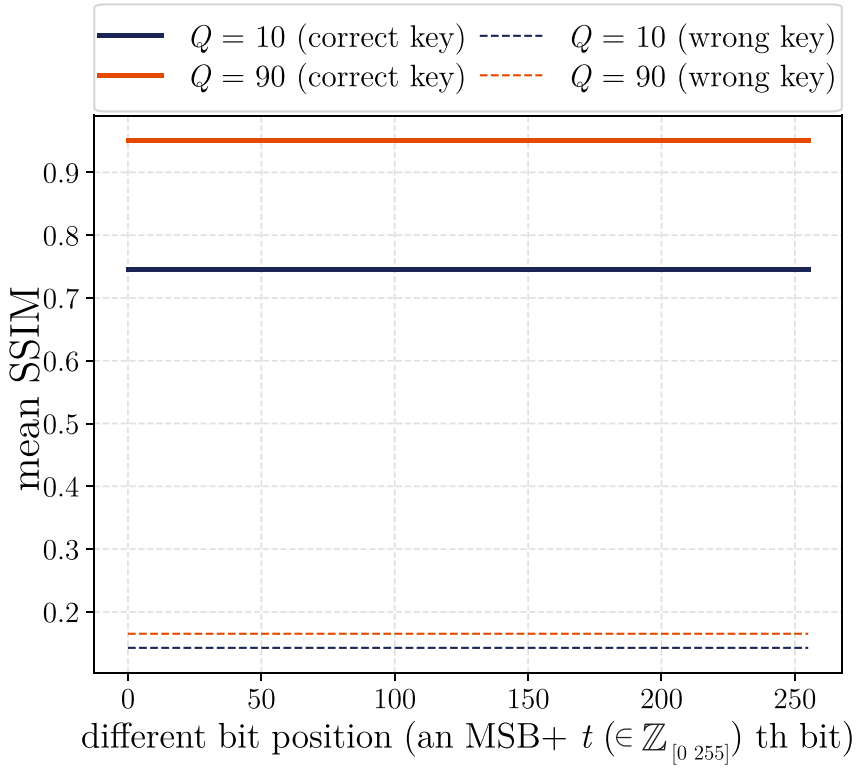


図 4.11　暗号化キーの各ビットが真正と異なっていた際の BEESB におけるキー感度解析.

よりも良好な符号化効率を示すものの，FIBS もまた調整可能性を実現困難である.

4.3.3　攻撃耐性

置換攻撃への頑健性

まず，暗号化デコード画像と元画像の間の平均 SSIM [67] を計測することで，置換攻撃への耐性を比較した．2^3 ビットキューブの場合，三種の m-キューブを用いた BEESB の攻撃耐性は低圧縮時に保たれているが，七種暗号化した場合は低圧縮になるほど，元画像の特徴を表す暗号化可能な m-キューブの増加によって頑健性が向上することがわかる（図 4.9(a)）．3^3 ビットキューブの場合，七種までの

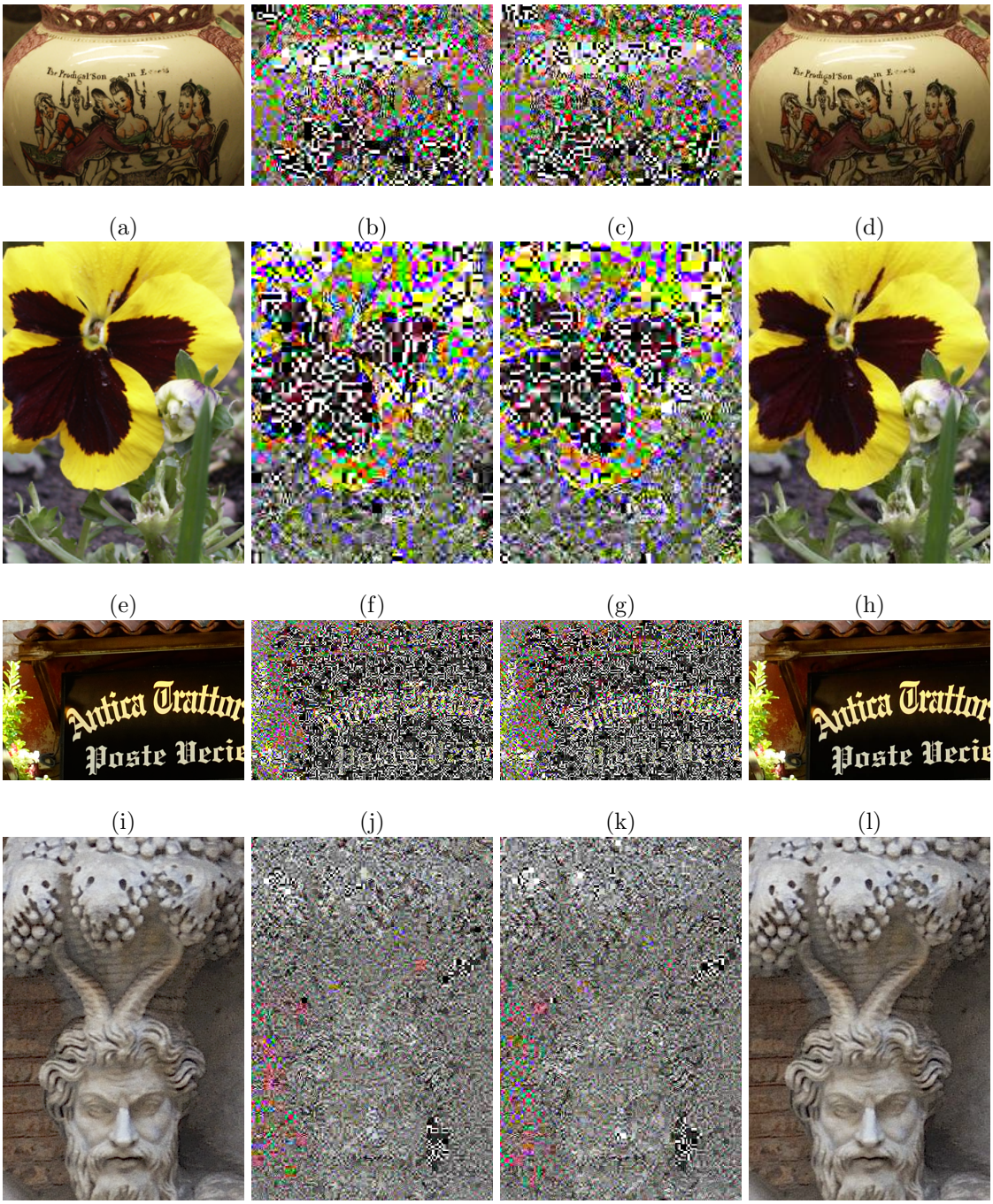


図 4.12　異なるキーを用いた際の攻撃耐性（JPEG $Q = 70$・$\ell^{[3]}_{1,\cdots,26} = 100$）：(a-d) *ucid00059* の一部, (e-h) *ucid00069* の一部, (i-l) *r00444b95t* の一部, (m-p) *r00b8d4a2t* の一部, (a,e,i,m) 元画像, (b,f,j,n) 真正のキー $\mathcal{F}_0$ と全く異なるキー $\mathcal{F}_1$ での復元結果, (c,g,k,o) 真正のキー $\mathcal{F}_0$ と LSB のみ異なるキー $\mathcal{F}_2$ での復元結果, (d,h,l,p) 真正のキー $\mathcal{F}_0$ での平文化結果.

m-キューブを用いた BEESB では頑健性が低いものの，13 種以上暗号化した場合から急激に向上することがわかる（図 4.9(b)）．4^3 ビットキューブの場合，26 種以上の m-キューブを暗号化した際の頑健性は充分に高いことがわかる（図 4.9(c)）．加えて，図 4.10 では BEESB（$\ell^{[2]}_{1,\cdots,7} = 100$）・DCACS

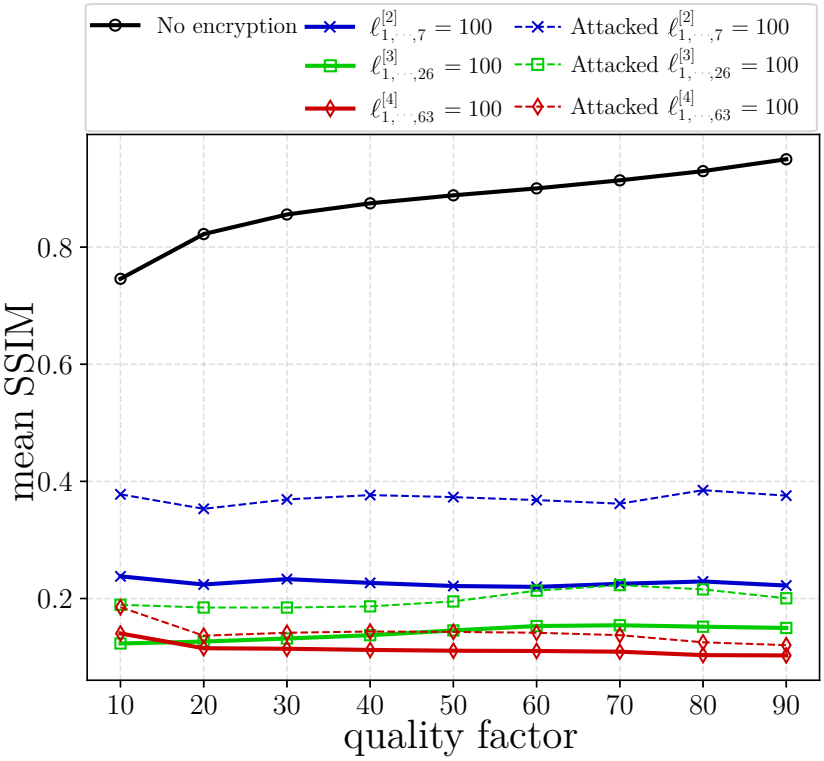


図 4.13　暗号化に用いた乱数列と非常に近い乱数列を用いた場合での復元品質.

表 4.6　SSIM による視覚品質の指標.

SSIM	指標
0.98+	indistinguishable from the original（元とほぼ同品質）
0.95	barely watchable（かろうじて視聴を許容可能）
0.9	really ugly（非常に低品質）
0.8	cannot still produce（元の品質とは言い難い）
0.7	fail to generate（本来生成不可能）

($\ell_{DC} = 7, \ell_{AC} = 5$)・RSF で暗号化した画像とその置換攻撃適用後の画像を，JPEG $Q = 50, 90$ の場合で示している．4.3.2 節で最もビットレート増加抑制を達成していると述べた DCACS・RSF では，置換攻撃によって知覚劣化を軽減されてしまうことがわかる．表 4.6 に示す ‘SSIM が 0.9 より小さな値である場合，劣化の品質は非常に酷い（really ugly）’ という指標 [69] により，SSIM < 0.5 の画像（図 4.9）は充分に酷い品質であるため，攻撃後も元の品質に復元されていないと考えられ，SVOD などでの一定レベルのセキュリティが期待される場面での使用が期待できる．さらに，パラメータ $\ell^{[2]}_{1,\cdots,7}, \ell^{[3]}_{1,\cdots,26}, \ell^{[4]}_{1,\cdots,63}$ による暗号化は低圧縮になるほど強固な保護を実現できるため，SNS などで高レベルのセキュリティが要求される場合に勧められる．

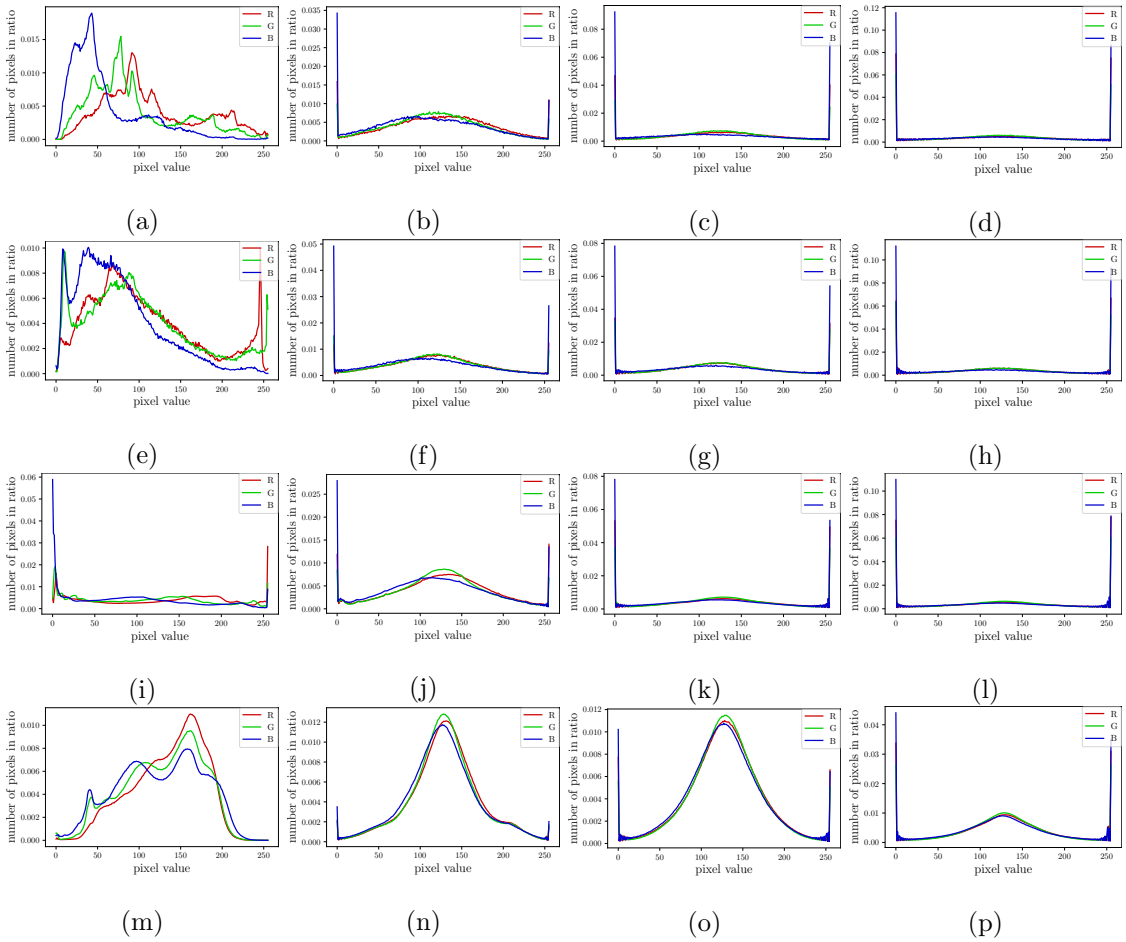


図 4.14　BEESB での暗号化デコード画像のヒストグラム解析 (JPEG $Q = 70$) : (a-d) *ucid00059*, (e-h) *ucid00069*, (i-l) *r00444b95t*, (m-p) *r00b8d4a2t*, (a,e,i,m) nonencrypted, (b,f,j,n) $\ell^{[2]}_{1,\cdots,7} = 100$, (c,g,k,o) $\ell^{[3]}_{1,\cdots,26} = 100$, (d,h,l,p) $\ell^{[4]}_{1,\cdots,63} = 100$.

総当たり攻撃への頑健性

続いて総当たり攻撃への耐性も解析した．4.2.3 節で述べたように，本研究では SHA-256 ハッシュ値 [52] を充分な長さの暗号化キーとして用いる．暗号化コンテンツが元の品質に偶然平文化されてしまう可能性があるため，真正のキーの各ビットが異なったキーを用いた際のキー感度解析を行った．図 4.11 に示すように，MSB から LSB にかけて，真正のものと異なるキーを用いた場合でも元品質に平文化できないことを確認した．また，主観的にも暗号化が真正のキー以外で平文化できないことを確認した（図 4.12）．さらに，本来あり得ないが，攻撃者がたとえ真正のキーから生成される乱数列と常に非常に近い乱数列を用いた場合での復元品質を解析した（図 4.13）．このように，真正の乱数列とつねに LSB のみ異なっている乱数列が用いられた場合でも，充分な頑健性を実現することを確認した．この頑健性を裏付ける証明のために，BEESB で暗号化される暗号化可能領域の大きさ（ビット数）を計測した．本実験では JPEG 品質ファクター Q が増加する（低圧縮になる）ほど，QDCT 係数領域での暗号化可能領域が増加することを確認した（表 4.7）．これは高い圧縮品質であるほど，QDCT 係

(a) (b) (c) (d)

図 4.15　映像符号化 HEVC への BEESB の適用結果：(a) 元映像 *Akiyo* の第一復号（I）フレーム，(b) BEESB（$\ell^{[2]}_{1,\cdots,7} = 100$），(c) BEESB（$\ell^{[3]}_{1,\cdots,26} = 100$），(d) BEESB（$\ell^{[2]}_{1,\cdots,63} = 100$）.

数を量子化で削減しない（保持する）ことで，より多くの非ゼロビットを BEESB で暗号化できるようになるためである．$\ell^{[4]}_{1,\cdots,63}$ による BEESB で最大量の暗号化可能ビットを暗号化したのは，4^3 ビットキューブが $2^3 \cdot 3^3$ ビットキューブよりも多くのゼロ／非ゼロビットを包含できることによる．これらの暗号化可能領域（ビット数）は 256 ビットの暗号化キー・19936 ビットの MT 状態空間ベクトル（次元数）よりも大きな数であるため，暗号化可能領域に基づいた鍵空間は元のキーに基づく鍵空間 2^{256}・MT 状態空間ベクトル次元数に基づく 2^{19936} よりも小さくならないことがわかる．それゆえに，より大きなビットキューブと多くのブロックを用いた BEESB が攻撃耐性を保証する．

ヒストグラム解析

また，BEESB による暗号化画像のヒストグラムを解析した結果を図 4.14 に示す．ビットキューブの辺幅を $h = 2, 3, 4$ のいずれから選択しても，暗号化画像のピクセル値は最大／最小値に漸近することが確認できる．さらに，ビットキューブを大きくするほど，ヒストグラム中央部の盛上がり部分が平坦になることがわかる．これらのことから，BEESB でのビットキューブサイズを大きくするほどより多くの色情報を秘匿化でき，適度な構造的情報も秘匿化できることがわかる．

4.4　映像符号化への応用

上記の BEESB は JPEG 符号化内の QDCT 係数領域に適用した場合での結果を示しているが，この領域での各 8×8 符号付十進数ブロックに適用できることから，同様の条件を満たす符号化領域内であれば映像符号化などにもプラグアンドプレイで適用できることがわかる．映像符号化規格 HEVC の量子化周波数係数領域に BEESB を適用した結果を図 4.15 に示す．本実験では，入力映像シーケンス *Akiyo* を HEVC の参照ソフトウェア HM [70] で符号化した際に，その CABAC エントロピー符号化直前の量子化周波数係数ブロックに BEESB を適用し，復号フレームの並び順は IPBPBPBPIPB$\cdots$ とした．暗号化後の映像シーケンスを平文化なしでデコードし，第一復号（I）フレームを抽出した．図 4.15 から，サイズ $2^3, 3^3, 4^3$ のビットキューブで HEVC の符号化ブロック（ビットキューボイド）を暗号化していることがわかる．2.1.2 節で述べたように，HEVC では JPEG のように符号化ブロック

を 8×8 に固定せず（領域ごとに 4×4 から 32×32 に変化させ），周波数変換にも DCT のみでなく DST も用いているが，これらに追従してどのサイズのブロックも暗号化していることがわかる．また，実験的に P,B フレームは予測によってほとんど残差信号からなるブロックとなったために暗号化による効果がほとんど現れず，I フレーム暗号化結果とほとんど差が見られなかったために本論文ではこれらの結果を割愛する．換言すると，暗号化 HEVC 圧縮映像のデコード時に，暗号化 I フレームの効果がフレーム内外予測で逆伝搬したことになり，I フレームの量子化周波数係数領域での暗号化がフレーム内のみならず P,B フレームにも影響（伝搬）することがわかる．また，AVC・VVC は HEVC と同様の構造をとっているため，AVC・VVC への BEESB の適用も自明にできる．このように，BEESB は画像符号化（JPEG）だけでなく映像符号化（AVC，HEVC，VVC）にも適用でき，汎用性を有する．

4.5 本章のまとめ

本章では新たな JPEG 符号化内フォーマット準拠暗号化であるビットキューボイドベース暗号化を提案した．ビットキューボイドベース暗号化は JPEG QDCT 係数領域の 8×8 ブロックを二進化することで得られるビットキューボイドの部分空間（ビットキューブ）を暗号化することによって，知覚劣化度合の柔軟な調整可能性を提供する．本章では同時に ESB を提案した．ESB は暗号化後の二値列を符号付十進数に「何の例外処理もいらず」再十進化可能な二進化であり，FSB で二進化後の負の二値列を一つ一つシフトすることによって，FSB の再十進化で要する例外処理を撤廃している．本研究では JPEG QDCT 係数領域で ESB によるビットキューボイドベース暗号化「BEESB」を適用した．JPEG 符号化内での暗号化を用いた実験では，暗号化 JPEG 圧縮画像の柔軟に調整可能な知覚劣化度合を実現し，ビットレートを従来よりも抑制し，種々の暗号文単独攻撃に高い耐性を持つことを示した．また，JPEG と同様の構造を持つ HEVC の符号化内に対しても BEESB を適用することで，画像符号化のみならず映像符号化にも汎用的なフォーマット準拠暗号化であることを示した．

表 4.7　BEESB における暗号化可能領域 [bits]（$\ell^{[2]}_{1,\cdots,7}$, $\ell^{[3]}_{1,\cdots,26}$, $\ell^{[4]}_{1,\cdots,63} = 100$ のそれぞれでの結果）.

画像	JPEG Q	BEESB		
		$\ell^{[2]}_{1,\cdots,7}$	$\ell^{[3]}_{1,\cdots,26}$	$\ell^{[4]}_{1,\cdots,63}$
ucid00001	50	96744	285120	427904
	70	125376	386208	520192
	90	246344	545670	702720
ucid00010	50	119744	340740	468032
	70	162520	436968	577216
	90	315600	624861	810368
ucid00020	50	125584	365283	440448
	70	175592	448335	591360
	90	337688	674271	868288
ucid00030	50	128632	339795	507328
	70	169880	439344	612416
	90	322304	620325	812544
ucid00040	50	77920	246348	356416
	70	101456	299538	453888
	90	167824	471852	601664
ucid00050	50	93752	305532	383808
	70	118512	360153	462272
	90	209056	527715	728000
ucid00060	50	82808	288225	370176
	70	109480	350730	599360
	90	200000	525906	702848
ucid00070	50	92848	302724	376320
	70	132840	374436	514624
	90	255200	559656	738176
ucid00080	50	134480	331236	474752
	70	175848	342171	574272
	90	342376	588708	797888
Mean	50	105154	315836	407042
	70	143589	392176	522859
	90	271216	580330	758704

第 5 章

おわりに

本研究では，画像・映像コンテンツのフォーマット準拠暗号化に存在する三つの社会的要求：再符号化可能性，調整可能性，汎用性を実現すべく，コンテンツの様々な領域での ‘直方体（キューボイド）’ を用いる二つのフォーマット準拠暗号化を提案した．

3 章では，まず一つ目に再符号化可能性を実現しつつ従来の暗号化手法よりも高セキュリティなピクセルキューボイドベース暗号化を提案した．ピクセルキューボイドベース暗号化は，非圧縮映像シーケンスを直方体（キューボイド）ベースで暗号化することにより，従来のピクセルブロックベース暗号化のもつ ‘非暗号化ブロック内のテクスチャを用いてブロックを結合させるジグソーパズル解読攻撃への脆弱性を撤廃しがたい’ という問題を解決しつつ，従来のピクセルブロックベース暗号化と同様にプロバイダ側でコンテンツの JPEG 再符号化が行われた場合でもコンテンツの符号化効率を維持する特性を持つ．

4 章では，二つ目にコンテンツの秘匿化（知覚劣化）度合の調整可能性を実現しつつ，頑健な攻撃耐性を維持するビットキューボイドベース暗号化を提案した．ビットキューボイドベース暗号化は，圧縮符号化内の符号付十進数を二進化した二進空間内をキューボイドベースで暗号化することにより，コンテンツの知覚劣化度合を柔軟に調整可能にすることで，SNS や SVOD などで要求される様々なセキュリティレベルに対応可能にした．また，この暗号化は総当たり攻撃・置換攻撃をはじめとした暗号化画像の暗号文単独攻撃に強固な耐性を持つことを示した．本研究では，提案の暗号化が JPEG QDCT 係数領域で適用できるということをまず示し，その性質上ほかの符号化に対しても，その符号化が「ブロックベースの変換・量子化」という同様の構造を取っていれば，画像符号化 JPEG のみならず映像符号化 AVC ／ HEVC などにも適用可能である，すなわち汎用性を持つことを示した．

また，本研究は画像・映像コンテンツへの汎用性を考慮した研究を行ったことを省み，今後の展望としてこれら以外のフォーマットへも汎用性を担保するフォーマット準拠暗号化の実現を目指す．たとえば点群データ符号化も画像・映像符号化に似た「非相関化・量子化」の構造をとっており，量子化後の非相関化領域であればビットキューボイドベース暗号化は点群を含む多様なフォーマットにも応用でき

る．ただし，符号化の変換（非相関化）・量子化部がブロックベースでない場合，ブロックによらない適用をするために元の手法を再設計する必要はある．フォーマット準拠暗号化があらゆるマルチメディアコンテンツの価値をセキュリティ面で今後さらに豊かにするであろうことを願い，本論文を結ぶ．

参考文献

[1] “簡単にわかる暗号の歴史,” White Paper Prepared For Symantec Corporation, 2014.

[2] C. E. Shannon, “Communication theory of secrecy systems,” *Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, Oct. 1949.

[3] National Institute of Standards and Technology, *FIPS PUB 197: Advanced Encryption Standard (AES)*. pub-NIST, Nov. 2001.

[4] 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.

[5] I.-T. R. T.800, *Information technology - JPEG 2000 image coding system: Core coding system*, 2019.

[6] I.-T. R. T.832, *Information technology - JPEG XR image coding system: Image coding specification*, 2019.

[7] I. J. WG1N83038, *JPEG White paper: JPEG XS, a new standard for visually lossless low-latency lightweight image coding system*, 2019.

[8] T. Richter, A. Artusi, and T. Ebrahimi, “JPEG XT: A new family of jpeg backward-compatible standards,” *IEEE Multimedia*, vol. 23, no. 3, pp. 80–88, July-Sept. 2016.

[9] P. Schelkens, T. Ebrahimi, A. Gilles, P. Gioia, K.-J. Oh, F. Pereira, C. Perra, and A. M. G. Pinheiro, “JPEG Pleno: Providing representation interoperability for holographic applications and devices,” *ETRI Journal*, vol. 41, no. 1, pp. 93–108, Feb. 2019.

[10] I.-T. R. H.222.0, *Information technology - generic coding of moving pictures and associated audio information: systems*, 1995.

[11] I.-T. R. H.264, *SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services – Coding of moving video — Advanced video coding for generic audiovisual services*, 2003.

[12] I.-T. R. H.265, *SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS Infrastructure of audiovisual services – Coding of moving video — High efficiency video coding*, 2013.

[13] B. Bross, J. Chen, and S. Liu, *Versatile Video Coding (Draft 2)*, JVET, 2018.

[14] "Twitter," https://twitter.com.

[15] "Facebook," https://www.facebook.com.

[16] "Hulu," https://www.hulu.jp.

[17] "Netflix," https://www.netflix.com/jp/.

[18] N. G. Bourbakis, "Image data compression-encryption using G-scan patterns," in *Proc. IC-SMC*, Orlando, FL, Oct. 1997, pp. 1117–1120.

[19] H. Cheng and X. Li, "Partial encryption of compressed images and videos," *IEEE Trans. Signal Process.*, vol. 48, no. 8, pp. 2439–2451, Aug. 2000.

[20] M. Ito, N. Ohnishi, A. Alfalou, and A. Mansour, "New image encryption and compression method based on independent component analysis," in *Proc. ICTTA*, Damascus, Syria, Apr. 2008, pp. 1–6.

[21] D. Maheswari and V. Radha, "Secure layer based compound image compression using XML compression," in *Proc. ICCIC*, Coimbatore, India, Dec. 2010, pp. 494–498.

[22] L. Tang, "Methods for encrypting and decrypting MPEG video data efficiently," in *Proc. ACMMM*, Boston, MA, Nov. 1996, pp. 219–229.

[23] B. Zeng, A. Yeung, S. Kei, S. Zhu, and M. Gabbouj, "Perceptual encryption of H.264 videos: Embedding sign-flips into the integer-based transforms," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 2, pp. 309–320, Feb. 2014.

[24] P. Li and K.-T. Lo, "Joint image compression and encryption based on order-8 alternating transforms," *J. Vis. Commun. Image*, vol. R, no. 44, pp. 61–71, Apr. 2017.

[25] A. Kingston, S. Colosimo, P. Campisi, and F. Autrusseau, "Lossless image compression and selective encryption using a discrete Radon transform," in *Proc. ICIP*, San Antonio, TX, Sep. 2007, pp. 465–468.

[26] F. Ahmed, M. Y. Siyal, and V. U. Abbas, "A perceptually scalable and JPEG compression tolerant image encryption scheme," in *Proc. PRIVT*, Singapore, Nov. 2010, pp. 232–238.

[27] K. Kurihara, M. Kikuchi, S. Imaizumi, S. Shiota, and H. Kiya, "An encryption-then-compression system for JPEG/Motion JPEG standard," *IEICE Trans. Fundamentals*, vol. E98-A, no. 11, pp. 2238–2245, Nov. 2015.

[28] H. Hofbauer, A. Uhl, and A. Unterweger, "Transparent encryption for HEVC using bit-stream-based selective coefficient sign encryption," in *Proc. ICASSP*, Florence, Italy, May 2014, pp. 1986–1990.

[29] D. Engel, T. Stütz, and A. Uhl, "A survey on JPEG2000 encryption," *Multimedia systems*, vol. 15, no. 4, pp. 243–270, Jan. 2009.

[30] G. Hong, C. Yuan, Y. Wang, and Y. Zhong, "A quality-controllable encryption for H.264/AVC video coding," in *Proc. PCM LNCS 4261*, Hangzhou, China, Nov. 2006, pp. 510–517.

[31] Y. Wang, M. O'Neill, and F. Kurugollu, "A tunable encryption scheme and analysis of fast selective encryption for CAVLC and CABAC in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 9, pp. 1476–1479, Sept. 2013.

[32] F. Peng, X. Zhang, Z.-X. Lin, and M. Long, "A tunable selective encryption scheme for H.265/HEVC based on chroma IPM and coefficient scrambling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 8, pp. 2765–2780, Aug. 2020.

[33] I.-T. R. T.81, *Information technology - digital compression and coding of continuous-tone still images - requirements and guidelines*, 1993.

[34] "JPEG privacy & security abstract and executive summary," https://jpeg.org/items/20150910_privacy_security_summary.html.

[35] K. Shimizu, T. Suzuki, and K. Kameyama, "Cube-based encryption-then-compression system for video sequences," *IEICE Trans. Fundamentals*, vol. E101-A, no. 11, pp. 1815–1822, Nov. 2018.

[36] ——, "Lapped cuboid-based perceptual encryption for Motion JPEG standard," in *Proc. APSIPA ASC*, Honolulu, Hawaii, Nov. 2018, pp. 2022–2026.

[37] T. Chuman, W. Sirichotedumrong, and H. Kiya, "Encryption-then-compression systems using grayscale-based image encryption for JPEG images," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 6, pp. 1515–1525, Nov. 2019.

[38] A. Unterweger, K. V. Ryckegem, D. Engel, and A. Uhl, "Building a post-compression region-of-interest encryption framework for existing video surveillance systems," *Multimedia Systems*, vol. 22, no. 5, pp. 617–639, Oct. 2016.

[39] K. Minemura, K. Wong, Q. Xiaojun, and T. Kiyoshi, "A scrambling framework for block transform compressed image," *Multimed. Tools. Appl.*, vol. 76, no. 5, pp. 6709–6729, Mar. 2017.

[40] V. Itier, P. Puteaux, and W. Puech, "Recompression of JPEG crypto-compressed images without a key," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 3, pp. 646–660, Mar. 2020.

[41] J. Ting, K. Wong, and S. Ong, "Format-compliant perceptual encryption method for JPEG XT," in *Proc. ICIP*, Taipei, Taiwan, Sept. 2019, pp. 4559–4563.

[42] Y. Mao and M. Wu, "A joint signal processing and cryptographic approach to multimedia encryption," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 2061–2075, July 2006.

[43] W. Li and Y. Yuan, "A leak and its remedy in JPEG image encryption," *Int. J. Comput. Math*, vol. 84, no. 9, pp. 1367–1378, Sept. 2007.

[44] M. I. Khan, V. Jeoti, and M. A. Khan, "Perceptual encryption of JPEG compressed images using DCT coefficients and splitting of DC coefficients into bitplanes," in *Proc. ICIAS*, Kuala Lumpur, Malaysia, June 2010, pp. 1–6.

[45] K. Rao and P. Yip, *Discrete cosine transform: algorithms, advantages, applications.* Academic Press, 1990.

[46] "Radiance - *Reference*," https://floyd.lbl.gov/radiance/framer.html.

[47] B. E. Bayer, "Color imaging array," US Patent 3 971 065, July 20, 1976.

[48] R. Durstenfeld, "Algorithm 235: Random permutation," *Commun. ACM*, vol. 7, no. 7, p. 420, 1964.

[49] B. Jun and P. Kocher, "The Intel random number generator," Cryptography Research, Inc. White Paper Prepared For Intel Corporation, Apr. 1999.

[50] M. Matsumoto and T. Nishimura, "Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Trans. Model. Comput. Simul.*, vol. 8, no. 1, pp. 3–30, Jan. 1998.

[51] M. Matsumoto, M. Saito, T. Nishimura, and M. Hagita, "Cryptmt3 stream cipher," *New Stream Cipher Designs. LNCS*, vol. 4986, pp. 7–19, 2008.

[52] National Institute of Standards and Technology, *FIPS PUB 180-4: Secure Hash Standard (SHS).* pub-NIST, Aug. 2015.

[53] E. Barker, *NIST Special Publication 800-57 Part 1, Revision 5.* NIST, 2020.

[54] D. Sholomon, O. David, and N. S. Netanyahu, "A genetic algorithm-based solver for very large jigsaw puzzles," in *Proc. CVPR*, Portland, OR, Jun. 2013, pp. 1767–1774.

[55] K. Shimizu, Q. Wang, and T. Suzuki, "Ac prediction error propagation-based encryption for texture protection of JPEG compressed images," in *Proc. PCS*, Bristol, UK, Sept. 2021, p. PP, accepted.

[56] C.-J. Lian, K.-F. Chen, H.-H. Chen, and L.-G. Chen, "Analysis and architecture design of block-coding engine for EBCOT in JPEG 2000," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 3, pp. 219–230, Dec. 2003.

[57] V. Sze and M. Budagavi, "High throughput CABAC entropy coding in HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1778–1791, Dec. 2012.

[58] K. Shimizu and T. Suzuki, "Cube-based encryption connected prior to Motion JPEG standard," in *Proc. APSIPA ASC*, Kuala Lumpur, Malaysia, Dec. 2017, pp. 1811–1814.

[59] T. Chuman, K. Kurihara, and H. Kiya, "On the security of block scrambling-based etc systems against extended jigsaw puzzle solver attacks," *IEICE Trans. Inf. & Syst.*, vol. E101-D, no. 1, pp. 37–44, Jan. 2018.

[60] "Xiph.org video test media [derf's collection]," https://media.xiph.org/video/derf/.

[61] K. Shimizu and T. Suzuki, "Flexibly-tunable bitcube-based perceptual encryption within JPEG compression," in *Proc. ICASSP*, Barcelona, Spain, May 2020.

[62] ——, "Finely tunable bitcuboid-based encryption with exception-free signed binarization for JPEG standard," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 4895–4908, Sep. 2021.

[63] H. T. Sencar and N. D. Memon, "Identification and recovery of JPEG files with missing fragments," in *Proc. the Ninth Annual DFRWS Conference*, Sept. 2009, pp. 88–98.

[64] G. Schaefer and M. Stich, "UCID: An uncompressed color image database," in *Proc. SPIE, Storage and Retrieval Methods and Applications for Multimedia*, vol. 5307, San Jose, CA, Jan. 2004, pp. 472–480.

[65] D.-T. Nguyen, C. Pasquini, V. Conotter, and G. Boato, "RAISE: a raw images dataset for digital image forensics," in *Proc. MMSys*, Mar. 2015, pp. 219–224.

[66] T. Richter, "thorfdbg/libjpeg: A complete implementation of 10918-1 (JPEG) comming from jpeg.org (the ISO group) with extensions for HDR standardized as 18477 (JPEG XT)," https://github.com/thorfdbg/libjpeg, 2019, last Access: 25/03/2020.

[67] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[68] G. Bjøntegaard, *Calculation of average PSNR differences between RD-curves*, VCEG-M33, 2001.

[69] "[x264-devel] Re: Defining video quality," https://mailman.videolan.org/pipermail/x264-devel/2006-December/002398.html, last Access: 25/03/2020.

[70] "High Efficiency Video Coding (HEVC)," https://hevc.hhi.fraunhofer.de.

謝辞

本論文を執筆するにあたり，筑波大学大学院に来てから研究に関する多大なご指導と日頃からの相談などに乗っていただいた鈴木大三先生に，まず何よりの感謝を申し上げさせて頂きます．また，大学院にきた当初二年間で所属させて頂いた適応情報処理研究室にて，研究室運営から研究指導まで多大なご支援を頂いた亀山啓輔先生にも感謝申し上げさせて頂きます．鈴木先生・亀山啓輔先生には博士論文の審査も行って頂き，同じく審査にご協力くださった徳永隆治先生・滝沢穂高先生・片岸一起先生にも深い感謝を申し上げます．本研究は日本学術振興会からの科研費に基づくものであり，斯様な資金援助を頂き研究を進められたことに心よりの感謝を申し上げます．さいごに，自分のこれまでの生活を支えてくださった両親，有益な見識を培わせてくれた諸関係者様方に心よりの感謝を申し上げます．本研究はJSPS 科研費 20J14599 の助成を受けたものです．