

JPEG 画像の Encryption-then-Compression システムに基づくソーシャル ネットワークサービスにおけるプライバシー保護

前川 貴大[†] 栗原 健太[†] 貴家 仁志[†]

[†] 首都大学東京 〒191-0065 東京都日野市旭ヶ丘 6-6

E-mail: †{maekawa-takahiro,kurihara-kenta}@ed.tmu.ac.jp, ††kiya@tmu.ac.jp

あらまし 従来のセキュリティを考慮した画像通信モデルでは、圧縮後に暗号化を行う Compression-then-Encryption システムが一般に使用される。一方で、プライバシー保護の観点から、その逆、すなわち圧縮を行う前にコンテンツを暗号化させる形式である、Encryption-then-Compression(ETC) システムが注目されている。その ETC システムにおける暗号化方式として、JPEG 圧縮の使用を前提としたブロックベースの暗号化が提案されている。しかし、その暗号化方式では、色差信号の間引きと補間の処理の影響によって、歪みが発生してしまう場合がある。そこで本稿では、歪みの発生理由を考察し、その歪みの除去方法を提案する。その結果、JPEG 圧縮に基づく ETC システムにおいて、任意の色差補間が使用可能となる。

キーワード ETC システム, 画像暗号化

Privacy Protection in Social Networking Services based on an Encryption-then-Compression System for the JPEG Standard

Takahiro MAEKAWA[†], Kenta KURIHARA[†], and Hitoshi KIYA[†]

[†] Faculty of System Design, Tokyo Metropolitan University 6-6, Asahigaoka, Hino-shi, 191-0065 Japan

E-mail: †{maekawa-takahiro,kurihara-kenta}@ed.tmu.ac.jp, ††kiya@tmu.ac.jp

Abstract Conventional secure image transmission is based on a Compression-then-Encryption system where image compression has to be conducted prior to image encryption. On the other hand, Encryption-then-Compression(ETC) systems where some image encryption can be conducted prior to compression, is expected for privacy protection. A block-based encryption scheme with the assumption of using JPEG compression has been proposed as an image encryption one for ETC systems. However, the image encryption scheme generates the block artifact on images due to the effect of subsampling chroma signals. In this article, we consider the reason why the artifact is generated and then propose a method to remove it. As a result, the method enables us to use any JPEG decoders in the ETC system.

Key words ETC system, Image encryption

1. ま え が き

近年、ソーシャルネットワーク (SNS) やクラウドコンピューティングのような、情報処理のための分散システムが広く、急速に普及してきている。そのような環境において、コンテンツは、帯域制限があり、かつ安全性が確立されないチャンネル上で通信される。このような背景から、安全な通信を行うために、データ圧縮とセキュリティを同時に考慮した研究が多く行われている。[1], [2].

SNS を利用してユーザーが通信を行う場合、画像などのコンテンツは圧縮された後に暗号化される、Compression-then-

Encryption システムを用いて通信されている。また SNS では、プロバイダが受信した画像を再圧縮することが知られている [3]。したがって、CTE システムでは、ユーザの像を SNS プロバイダに開示しなければならないという問題が発生する。このことは、プライバシー保護は、プロバイダーの信頼性の仮定の下でのみ成り立つことを意味する。SNS での人為的なミスや事故に対しては、無防備な現状である。

そのような問題に対し、プライバシー保護の観点から、暗号化後に圧縮を行う Encryption-then-Compression(ETC) システムの使用が注目されている。[4]~[7]。ETC システムでは、暗号化鍵をユーザ自身が管理しており、プライバシー保護をプ

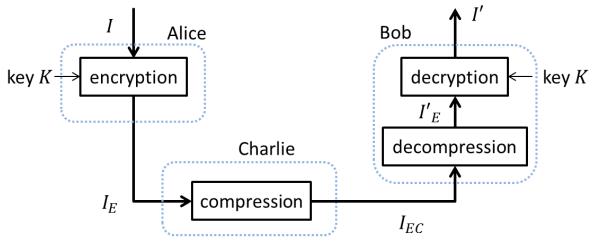


図1 Encryption-then-Compression システム

ロバイバーの信頼性を仮定せずに達成することができる。ETCシステム実現のために、圧縮可能な画像暗号化や、その安全性について研究されている [5]～[14]。本稿では、SNS への応用で特に有用な JPEG 圧縮方式の使用を前提とした暗号化方式に焦点をあてる。この方式は、暗号化後に JPEG 圧縮が可能であるが、その圧縮・伸長の条件によっては、ブロック歪が発生してしまう。本稿では、その歪み発生を理由を考察し、歪みを含んだ画像から歪みを除去する方法を提案する。

2. 準備

2.1 ETC システムおよび画像の知覚暗号化

一般に、画像コンテンツの通信モデルは、画像を圧縮した後に暗号化を施して、通信を行う、いわゆる Compression-then-Encryption(ETC) システムである。一方で、本稿では図1で示す通り、画像を暗号化したあとに圧縮をする Encryption-then-Compression システムに着目する。図1では Alice が信用できない通信チャンネルプロバイダの Charlie を通じて Bob にある画像 I を伝送することを想定している。

ETC システムにおいて、それらの暗号化は AES や RSA といった整数論理的な暗号化を用いた場合、一般にその暗号化データを圧縮することは困難である。さらに、非圧縮による歪みの発生も許容できない。そこで、画像を視覚的に暗号化する知覚暗号化の使用が期待されている。知覚暗号化では、暗号化後も画像の形式を保持しているため、処理が軽いことに加え、号化後にも画像圧縮などの処理が可能であるという特徴がある。さらに知覚暗号化は、AES などの暗号化手法と組み合わせて使用することもできる。知覚暗号化の使用により、プライバシーを保護した形式で柔軟に伝送およびコンテンツの活用が可能となる。

静止画像に対する知覚暗号化法として、サイズ $X \times Y$ の画像を一定サイズ $B_x \times B_y$ に分割して処理を行うブロックベース暗号化がある。[11]～[14](図2)。JPEG 圧縮を前提としたブロックベース暗号化 [11], [13] は以下の4ステップで行われる。

STEP1. ブロックスクランブル

ブロックスクランブルは、分割されたブロックを乱数を用いてランダムに置換する方法である。ブロックの分割とスクランブルの例を図3に示す。ブロック毎の位置情報を原画像のそれとは変更することにより、原画像の視認性を制御することを可能とする。ブロックサイズ $B_x \times B_y$ は色間引きと圧縮効率を考慮して、 16×16 画素が一般的に選択される。

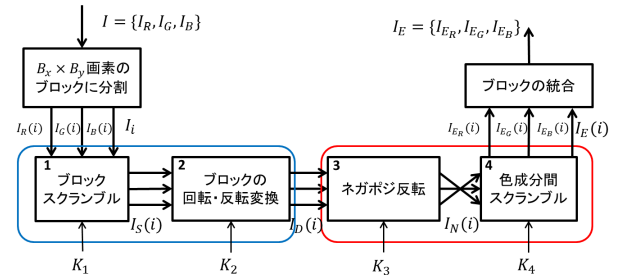


図2 ブロックベース暗号化のフロー

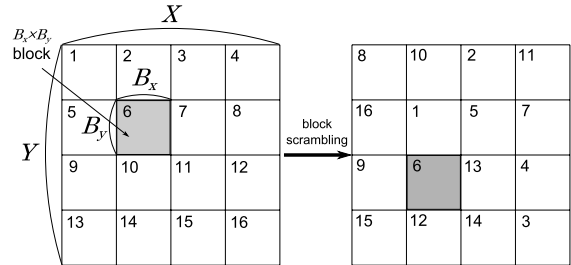


図3 $B_x \times B_y$ 画素のブロックへの分割とブロックスクランブルの例

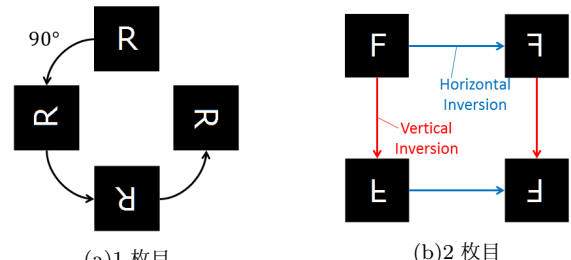


図4 ブロックの回転、反転のイメージ

STEP2. ブロックの回転、反転変換

ブロックの回転反転変換は、ブロックの位置関係を変更せずに、図4(a)に示すようにブロックを 0° , 90° , 180° , 270° のうちいずれかの角度だけランダムに回転させる方法である。一方ブロックの反転変換は、ブロックの位置関係を変更せずに、図4(b)に示すようにブロックを水平・垂直方向にランダムに反転させる方法であり、反転しない、もしくは水平・垂直方向どちらにも反転するといったことも起こり得る。

STEP3. ネガポジ反転

ブロックスクランブルや回転、反転変換は、ブロックの置換や幾何変換を行う方法であるため、その暗号化画像は原画像から色変換がなされていない。さらにヒストグラムは原画像のそれと同一出ある。暗号化画像からの原画像の特定を困難にするには、暗号化画像において色成分を変換すべきであると考えられる。そこで、色変換の一手段としてブロック毎に画素をランダムに反転させる、ネガポジ反転を使用する。

STEP4. 色成分間スクランブル

さらに画素の色成分を変換する方法として、カラー画像の R, G, B 成分の値をブロック内でランダムに置換する、色成分スクランブルを考える。ただし、色成分スクランブルにおけるブロックサイズを $B_x \times B_y$ とする。0 から 4 の 5 値のランダムな整数をブロックごとに割り当て、割り当てられた乱数に応じて

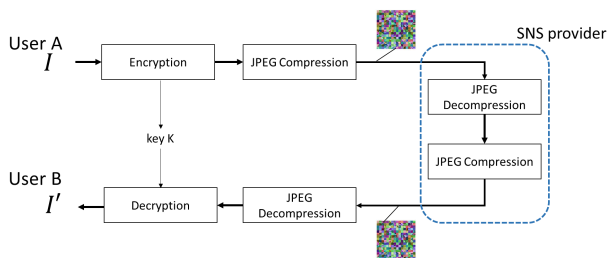


図 5 ETC システムの SNS への応用

ブロック内すべての画素の色成分を交換するという方法である。

2.2 SNS での応用

ソーシャルネットワーキング (SNS) における ETC システムの通信モデルを図 5 に示す。同図は UserA から複数人の限定された UserB に、SNS を介して画像を送信する際の通信である。UserA は SNS provider に圧縮した (または非圧縮の) 暗号化コンテンツを送信する。SNS provider は受け取ったコンテンツをファイルサイズ制御のために再圧縮する。ここで SNS provider は、暗号化されたコンテンツを受け取っており、かつ暗号化を解く鍵は所持していない。そのため SNS provider はコンテンツの内容を知ることができない。SNS provider から圧縮画像を受け取った UserB は、画像を伸長し、次に所有している鍵を用いてコンテンツを閲覧可能とする。

一方、Compression then Encryption (CTE) システムの場合、再圧縮を可能とするために SNS provider は鍵を所有する必要があり、その結果 SNS provider にコンテンツの内容が知られてしまう。そのため SNS の信頼性を仮定し、従来通信が行われてきた。しかし万が一の事故や人為的なミスなどで、SNS Provider からコンテンツの内容が漏れてしまう危険性がある。ETC システムでは鍵を SNS provider と独立して送信者と受信者との間で管理しており、ユーザーのプライバシーはそのような場合でも保護される。

2.3 ETC システムにおける歪みの発生

画像に対する JPEG 圧縮では、まず RGB 空間から輝度・色差 (YCbCr) 空間への変換が行われる。次に、CbCr 成分に対して間引きを行うことが許されている。ここで間引きタイプとして、 $[4:2:0]$ $[4:2:2]$ $[4:4:4]$ の三種類から選択される。 $[4:2:0]$ では、水平・垂直方向にそれぞれ $1/2$ の間引きを行う。 $[4:2:2]$ では、水平方向のみ $1/2$ の間引きを行う。 $[4:4:4]$ では間引きを行わない。JPEG 画像の伸長では、 $[4:2:0]$ または $[4:2:2]$ の間引きのタイプが選択された場合、逆に間引かれた CbCr 成分 (ここでは C'bC'r 成分) に対して補間処理が施される。

従って、その場合に、この C'bC'r 成分に施される補間処理に起因して、歪みが発生する可能性がある。一般に以下の 2 つの条件を同時に満たす伸長処理が行われたときに歪みが発生する。

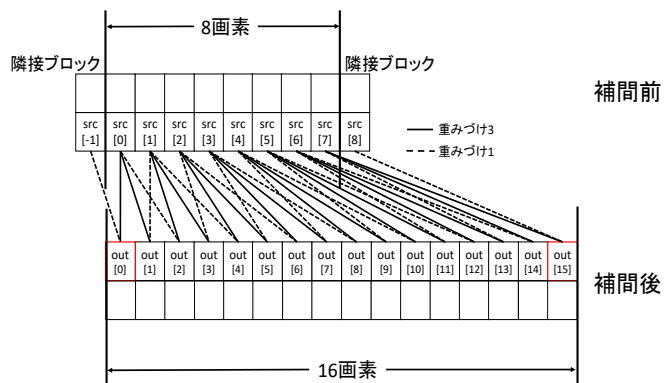


図 6 library JPEG での色差補間

- [条件 1]
 - 色差間引きが行われている。
 - ($[4:2:0]$ もしくは $[4:2:2]$ が選択)
- [条件 2]
 - 色差補間法が隣接ブロック画素を用いて行われている。

この補間処理は 8×8 のブロック内のデータのみを用いて行う方法や、隣接ブロックの画素との関係性を用いて行うものといった様々な方法が存在する。隣接するブロック間の関係性を用いて補間を行う場合、隣接ブロックの画素の影響を受けた新しい画素が補間によって生成されることになる。先の暗号化法では、一般に隣のブロックには実際のブロックとは異なるブロックが位置している。そのため 16×16 画素のブロック状の歪みが生じることになる

3. 提案法

3.1 補間による歪みの発生メカニズム

$B_x \times B_y = 16 \times 16$ 画素のブロックベース暗号化画像の、RGB 成分と同様に、YCbCr 成分もそれぞれ 16×16 画素のブロックベース単位で暗号化が施されることになる。代表的な JPEG 圧縮プログラム (library JPEG) [15] を例にして、歪み発生メカニズムを考察する。

library JPEG では圧縮時に色差間引きを実行した場合、伸長時に隣接ブロックの画素を用いて補間が実行される。図 6 で、水平方向の補間を説明している。ここで $src[x]$ は補間前の画素であり、 $out[x]$ は補間処理後の画素である。12 画素を用いて 16 画素が補間によって生成される各画素の関係は次式で示される。

$$\begin{cases} out[7] = \frac{1}{4}(src[4] + 3 \times src[3]) \\ out[6] = \frac{1}{4}(src[2] + 3 \times src[3]) \\ out[5] = \frac{1}{4}(src[3] + 3 \times src[2]) \\ out[4] = \frac{1}{4}(src[1] + 3 \times src[2]) \\ out[3] = \frac{1}{4}(src[2] + 3 \times src[1]) \\ out[2] = \frac{1}{4}(src[0] + 3 \times src[1]) \\ out[1] = \frac{1}{4}(src[1] + 3 \times src[0]) \\ out[0] = \frac{1}{4}(src[-1] + 3 \times src[0]) \end{cases} \quad (1)$$

$$\begin{cases} out[15] = \frac{1}{4}(src[8] + 3 \times src[7]) \\ out[14] = \frac{1}{4}(src[6] + 3 \times src[7]) \\ out[13] = \frac{1}{4}(src[7] + 3 \times src[6]) \\ out[12] = \frac{1}{4}(src[5] + 3 \times src[6]) \\ out[11] = \frac{1}{4}(src[6] + 3 \times src[5]) \\ out[10] = \frac{1}{4}(src[4] + 3 \times src[5]) \\ out[9] = \frac{1}{4}(src[5] + 3 \times src[4]) \\ out[8] = \frac{1}{4}(src[3] + 3 \times src[4]) \end{cases} \quad (2)$$

src[-1] と src[8] は隣接するブロックの画素である。今隣接するブロックには原画像のものとは異なるものが位置しているため、算出された out[0] と out[15] は暗号化の影響を受け、歪みを含むこととなる。これを同様に垂直方向にも補間を行うと、16 × 16 画素のブロックベース上に歪みが生じる。その後、YCbCr 空間から RGB 空間への変換が行われる。

次に、暗号の復号処理が行われる。ブロックベース暗号化では STEP4 で色成分間スクランブルが施されている。そのため、CbCr 成分が歪みを含む画素では、RGB 空間でも歪みを含むことになる。従って、暗号を復号した後に RGB 空間から YCbCr 空間に変換を行うと、Y、Cb、Cr それぞれが歪みを含むことになる。

3.2 歪みの除去

一般に、先に述べたように、暗号化を復元した画像では、暗号化に起因した歪みが、その YCbCr 成分にそれぞれ含まれる。その歪みの除去を次に考察する。

以下が提案する歪み除去の実行手順である。

ステップ 1. 暗号化領域に戻す

一度歪みを含んだ画像を暗号化鍵を用いて、暗号化領域に戻す。暗号化された状態では暗号化画像の Y 成分は歪みを含んでいない。そのため、CbCr 成分の歪みを除去、すなわち out[0] と out[15] の値を求め直すことが課題となる。

ステップ 2. 補間前の CbCr 成分を計算。

式 (1) および式 (2) は次式のように行列で表記される。

$$\begin{pmatrix} out[0] \\ out[1] \\ out[2] \\ out[3] \\ out[4] \\ \vdots \\ out[11] \\ out[12] \\ out[13] \\ out[14] \\ out[15] \end{pmatrix} = \begin{pmatrix} 1 & 3 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 3 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 1 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 3 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 1 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 3 & 1 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & 3 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 3 & 1 \end{pmatrix} \begin{pmatrix} src[-1] \\ src[0] \\ src[1] \\ src[2] \\ src[3] \\ \vdots \\ src[5] \\ src[6] \\ src[7] \\ src[8] \\ src[9] \end{pmatrix} \quad (3)$$

この逆の計算、すなわち out[n] から src[n] を求めることによって、補間前の C'bC'r 値を知ることができる。例えば、src[1] は

$$src[1] = \frac{1}{2}(3 \times out[2] - out[1]) \quad (4)$$

と与えられる。他の src[n] もほぼ同様であるが、着目するブロック内の値 src[0] から src[7] は、歪みを含んでいる画素 out[0]、out[15] を用いずに計算することが可能である。

ステップ 3. 再補間を実行する (Y 成分の歪み除去)

暗号化が施された状況では、隣接ブロックに原画像とは異なる場所のブロックが位置している。そのため、隣接ブロックを用いて補間を実行すると out[0] と out[15] に歪みを含むことになる。従って、本稿では、0 次ホールド処理

$$\begin{aligned} out[0] &= src[0] \\ out[15] &= src[7] \end{aligned} \quad (5)$$

を実行する。out[1] から out[14] については、歪みを含んでいないので再補間する必要はない。その後、暗号化の復号処理を行い、復号画像の Y を求める。

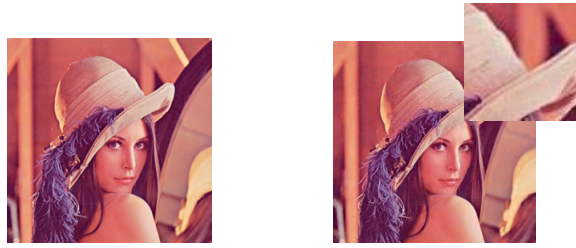
ステップ 4. CbCr 成分の再計算 (CbCr 成分の歪み除去)

暗号化が復号された状況では、隣接ブロックには原画像と同じブロックが位置している。従って、復号された画像の CbCr から補間前の C'bC'r を求めて、再度補間を実行することで歪みのない CbCr 成分を求めることができる。その際の補間は、式 (1) および式 (2) を直接使用できる。以上の手順によって、歪みのない YCbCr が生成され、それらから歪みのない RGB 画像を求めることができる。

4. シミュレーション

4.1 実験条件

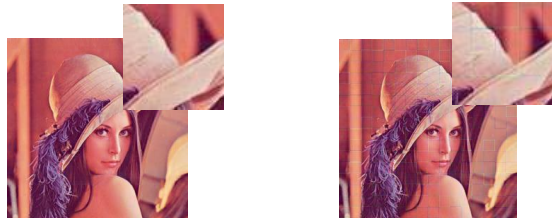
シミュレーションには、標準画像データベース SIDBA の 4 枚の画像 (Airplane, Girl, Lenna, Mandrill256 × 256) を用いた。JPEG 圧縮のプログラムは Joint Photographic Experts Group より library JPEG を使用した。さらに異なる色差補間 (ブロック内補間) を実行可能なプログラムとして、Independent JPEG Group より JPEG(9b) を入手して使用した。暗号化および色差信号の処理に着目した圧縮・伸長条件による画像分類を表 1 に示す。



(a) 原画像

(b) I_2

(歪みなし, PSNR : 32.45[dB])

(c) I_3 (d) I_4

(歪みなし, PSNR : 32.12[dB])

(歪みあり, 28.35[dB])

図 7 歪みの発生条件 (PSNR)

表 1 圧縮・伸長条件による画像分類

(* : library JPEG, ** : JPEG(9b))

	I_1^*	I_2^*	I_3^{**}	I_4^*
暗号化	あり	なし	あり	あり
色差間引き	[4 : 4 : 4]	[4 : 2 : 0]	[4 : 2 : 0]	[4 : 2 : 0]
色差補間法	必要なし	隣接ブロック 使用	ブロック内	隣接ブロック 使用

4.2 歪みの発生の有無

図 7 に同一画像に対して、表 1 に示した条件の下で圧縮・伸長した結果を同図 (b)-(d) に示す。ここでは、JPEG 圧縮後の画像は 1.53 [bpp] である。先に述べたように、同図 (d) のみ暗号化の影響を受けた歪みが発生する。提案法の目的は、SNS などの圧縮・伸長条件をユーザーが制御できない条件下で発生したこのような歪みを、除去するための方法を与えることにある。

4.3 提案法による歪みの除去

$B_x \times B_y = 16 \times 16$ 画素のブロックベース暗号化が施された暗号化画像を復号した際に生じる歪みを含んだ画像に対して、提案法を適用する。4 枚の画像について、それぞれ以下の 1)~5) の手順を実行する。

- 1) 画像に知覚暗号化を施す。
- 2) 暗号化画像を JPEG 形式で圧縮/伸長
- 3) 知覚暗号化を復号
- 4) 提案法による歪みの除去
- 5) PSNR の計算

表 1 に示した条件下での結果を表 2 に示す。ここで、 I_5 は、歪みを含む I_4 の画像に対して提案法を適用した場合の結果である。 I_4 と I_5 に着目すると、平均で 1.82 [dB] の PSNR の向上が確認できる。提案法を用いて歪みを除去した画像を図 8 に示

表 2 伸長された各画像の PSNR [dB]

(○ : 直接適可, × : 歪みあり (要除去), ● : 歪み除去後)

	ビットレート [bpp]	I_1	I_2	I_3	I_4	I_5
Airplane	1.59	31.94	30.36	30.21	29.43	30.14
Girl	1.20	33.14	32.80	32.44	31.20	32.60
Lena	1.54	32.62	32.45	32.12	28.35	32.30
Mandrill	2.43	27.14	27.25	27.05	25.93	27.15
平均	1.69	31.21	30.71	30.68	28.73	30.55
ETC システムへの適合性		○	-	○	×	●

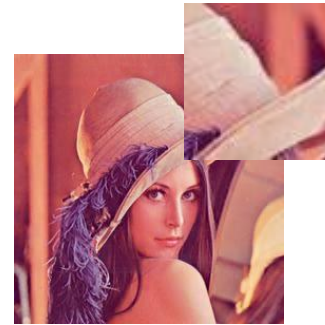


図 8 歪みを除去した画像

PSNR : 32.30 [dB]

す。同図では、歪みが除去されており、視覚的にも自然な画像が復元されていることが確認できる。

5. まとめ

本稿では、JPEG 圧縮を想定した Encryption-then-Compression システムにおける暗号化領域での色差補間に起因した歪み除去する手法を提案した。歪みを含まない色差成分の補間前の値を求めることにより、歪みを取り除く手法である。シミュレーションにより、視覚的な歪みの除去、また歪みを除去したことによる、PSNR の向上を確認した。ブロックベースの暗号化を利用した ETC システムでは、これまで国際標準符号化の JPEG 圧縮における、使用できる色差補間法に制限があった。提案法によって、その制限が原理的に解除され、圧縮時に色差間引きを行った場合でも、任意の色差補間法の使用が可能となった。

文 献

- [1] C. T. Huang, L. Huang, Z. Qin, H. Yuan, L. Zhou, V. Varadharajan, and C. C. J. Kuo, "Survey on securing data storage in the cloud," APSIPA Transactions on Signal and Information Processing, vol.3, e7, Jun. 2014.
- [2] R. L. Lagendijk, Z. Erkin, and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," IEEE Signal Processing Mag., vol. 30, no. 1, pp. 82-105, Jan. 2013.
- [3] A. Viejo, J. Castella-Roca and G. Rufian, "Preserving the User's Privacy in Social Networking Site," In Proceedings of Trust, Privacy, and Security in Digital Business: 10th International Conference, pp. 62-73, Aug. 2013.
- [4] M. Johnson, P. Ishwar, V. Prabhakaran, D. Schonberg, and K. Ramchandran, "On compressing encrypted data," IEEE

Trans. Signal Processing, vol.53, no.10, pp.2992-3106, Oct. 2004.

- [5] J. Zhou, X. Liu, O. C. Au, and Y. Y. Tang, "Designing an Efficient Image Encryption-then-Compression System via Prediction Error Clustering and Random Permutation," IEEE Transactions on Information and Forensics and Security, vol.9, no.1, pp.39-50, Jan. 2014.
- [6] R. Hu, X. Li, and, B. Yang, "A new lossy compression scheme for encrypted gray-scale images," In Proceeding of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp.7436-7440, May 2014
- [7] X. Zhang, "Lossy compression and iterative reconstruction for encrypted image," IEEE Trans. Inf. Forens. Security, vol.6, no.1, pp.53-58, Mar. 2011.
- [8] T. Chuman, K. Kurihara, and H. Kiya, "On the Security of Block Scrambling-based ETC Systems against Jigsaw Puzzle Solver Attacks," Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, New Orleans, LA, USA, 5th March, 2017.
- [9] 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," International Conference on Intelligent and Advanced Systems (ICIAS), pp. 1-6, Jun. 2010.
- [10] O. Watanabe, A. Uchida, T. Fukuhara, and H. Kiya, "An encryption-then-Compression System for JPEG 2000 Standard," In Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp.1226-1230, Apr. 2015.
- [11] K. Kurihara, S. Shiota, and H. Kiya, "An Encryption-ThenCompression System for JPEG Standard," In Proceedings of Picture Coding Symposium 2015, pp.119-123, June 2015.
- [12] K. Kurihara, S. Imaizumi, S. Shiota, and H. Kiya, "An Encryption-then-Compression System for Lossless Image Compression Standards," IEICE Trans. Inf. & Sys., vol.E100-D, no.1, pp.52-56, January 2017.
- [13] K. Kurihara, M. Kikuchi, S. Imaizumi, S. Shiota, and H. Kiya, "An Encryption-then-Compression System for JPEG/motion JPEG standard," IEICE Transactions on Fundamentals, vol.E98-A, no.11, pp.2238-2245, Nov. 2015.
- [14] K. Kurihara, O. Watanabe, and H. Kiya, "An EncryptionThen-Compression System for JPEG XR Standard," In Proceedings of the IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), Jun. 2016.
- [15] <https://jpeg.org/>