



HAL
open science

JPEG Steganography and Synchronization of DCT Coefficients for a Given Development Pipeline

Théo Taburet, Patrick Bas, Wadih Sawaya, Rémi Cогranne

► **To cite this version:**

Théo Taburet, Patrick Bas, Wadih Sawaya, Rémi Cогranne. JPEG Steganography and Synchronization of DCT Coefficients for a Given Development Pipeline. ACM Workshop on Information Hiding and Multimedia Security, Jun 2020, Denver, United States. 10.1145/3369412.3395074 . hal-02553023v2

HAL Id: hal-02553023

<https://hal.science/hal-02553023v2>

Submitted on 22 Sep 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

JPEG Steganography and Synchronization of DCT Coefficients for a Given Development Pipeline

Théo Taburet

theo.taburet@centralelille.fr

Univ. Lille, CNRS, Centrale Lille, UMR 9189, CRISTAL
Lille, France

Wadih Sawaya

wadih.sawaya@imt-lille-douai.fr

IMT Lille-Douais, Univ. Lille, CNRS, CRISTAL
Lille, France

Patrick Bas

Patrick.Bas@centralelille.fr

Univ. Lille, CNRS, Centrale Lille, UMR 9189, CRISTAL
Lille, France

Rémi Cogranne

remi.cogranne@utt.fr

LM2S Lab. - ROSAS Dept., Troyes Univ. of Technology
Troyes, France

ABSTRACT

This paper proposes to use the statistical analysis of the correlations between DCT coefficients to design a new synchronization strategy that can be used for cost-based steganographic schemes in the JPEG domain. First, an analysis applied on the photonic noise is performed on the covariance matrix of DCT coefficients of neighboring blocks after a development pipeline similar to the one used to generate BossBase. This analysis exhibits (i) a decomposition into 8 disjoint sets of uncorrelated coefficients (4 sets per block used by 2 disjoint lattices) and (ii) the fact that each DCT coefficient is correlated with 38 other coefficients belonging either to the same block or to connected blocks. Using the uncorrelated groups, an embedding scheme can be designed using only 8 disjoint lattices. The proposed embedding scheme relies on the following ingredients. Firstly, we convert the empirical costs associated to one each coefficient into a Gaussian distribution whose variance is directly computed from the embedding costs. Secondly we derive conditional Gaussian distributions from a multivariate distribution considering only the correlated coefficients which have been already modified by the embedding scheme. This covariance matrix takes into account both the correlations exhibited by the analysis of the covariance matrix and the variance derived from the costs. This synchronization scheme enables to obtain a gain of P_E of at least 7% at $QF95$ for an embedding rate close to 0.3 bnzac coefficient using DCTR feature sets for both UERD and J-Uniward.

CCS CONCEPTS

• **Security and privacy** → *Domain-specific security and privacy architectures*; **Intrusion/anomaly detection and malware mitigation**; **Malware and its mitigation**;

KEYWORDS

Digital image steganography, JPEG domain, sensor noise, image processing pipeline, covariance

ACM Reference Format:

Théo Taburet, Patrick Bas, Wadih Sawaya, and Rémi Cogranne. 2020. JPEG Steganography and Synchronization of DCT Coefficients for a Given Development Pipeline. In *Proceedings of -*. ACM, New York, NY, USA, 11 pages. <https://doi.org/10.1145/nnnnnnn.nnnnnnn>

1 INTRODUCTION

1.1 Previous works

In order to increase the practical security of steganographic algorithms for digital images, one strategy is to synchronize embedding changes on samples that are correlated. The dependencies between image samples can come from correlations within the Cover contents, for example on homogeneous areas or textures, or correlations induced by the development pipeline (downscaling [2], demosaicking [13], DCT transforms [13],...).

The synchronization process is, however, difficult to implement since this process is antagonist with the general principle of additive distortion commonly used in steganography [7] which considers independent embedding changes and which is practically achieved using Syndrome Trellis Codes [7].

One common strategy to deal with this issue is to break the dependencies by decomposing the set of image coefficients (pixels or DCT coefficients) into sets of disjoint lattices.

Existing synchronization methods can be divided into two categories depending on whether synchronization is carried out on the cost-map or on the embedding probabilities.

Synchronization of the cost map: The first scheme to propose synchronizing the cost map is based on Gibbs sampling, and it was proposed by Filler *et al.* [6] and improved by Denemark *et al.* [5] with the "synch" implementation. The proposed stego scheme works in the spatial domain and uses two lattices associated with a chessboard-like geometry: once the embedding is performed in the first lattice, the costs are then adjusted in the second one so that consistent local modification changes are more likely to be performed. Independently, a very similar idea was proposed by Li *et al.* [11] using four lattices, but without performing multiple sweeps through the lattices (actually the analysis in [5] shows that only one sweep is necessary to maximize the performance, so the two strategies are very similar).

Synchronization of embedding probabilities: The other class of synchronization schemes proposes to modify the embedding

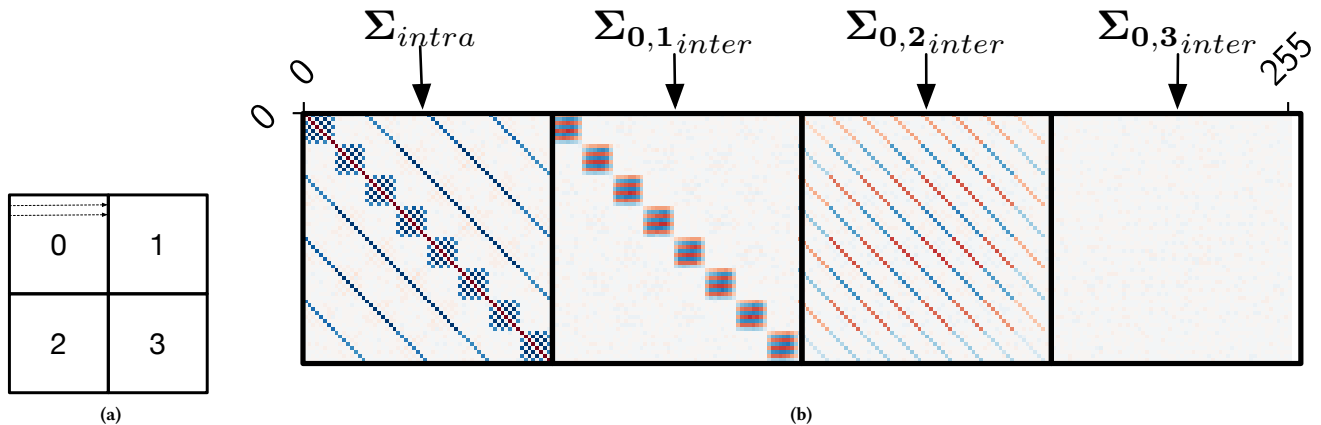


Figure 1: (a) : scan order by block and coefficients, (b) Intra and inter correlations exhibited by the correlation matrix \hat{R} . Blue colors denote negative correlation coefficients. See also the Appendices for the list of correlated DCT modes.

probabilities directly, these schemes are dedicated to Natural Steganography proposed by Bas *et al.* [1], where the stego signal tries to mimic the sensor photonic noise. In order to maximize the practical security after down-sampling in the spatial domain [2] or after demosaicking in the JPEG domain [13], the multivariate distribution of the stego signal is decomposed into conditional distributions over disjoint lattices using the chain rule of conditional probabilities. On a given lattice, the stego signal can be generated independently (conditionally to the embedding performed on the previous lattices) and a classical STC can be used.

Between these two classes there exist hybrid strategies proposed by Zhang *et al.* [14] and Li *et al.* [12] that define joint costs between samples and then derive a joint probability which is after decomposed into conditional probabilities and costs.

In the JPEG domain, as far as we know the only schemes addressing this issues are proposed by Li *et al.* [12] and Taburet *et al.* [13]. Even if these two schemes use completely different rationales and rely on completely different embedding schemes, they both try to preserve the continuities between adjacent JPEG blocks during embedding.

1.2 Main ideas

The present paper proposes a novel method that combines the advantages of both prior works [12, 13]. On one hand, the method can be easily applied in practice in the sense that, as proposed in [12], we use a cost map derived from a classical JPEG embedding scheme such as UERD [8] or J-Uniward [10]. On the other hand, the main contribution of the proposed method relies on its statistically-based foundation since, as in [13], it exploits the correlations induced by the development pipeline to synchronize the embedding changes. However, contrary to [13], the proposed synchronization method can be applied with any cost based steganographic scheme. The main idea proposed in this paper is to leverage the natural correlations induced by the development pipeline on the photonic noise to perform synchronization in the JPEG domain.

We first analyze the covariance matrix applied on the photonic noise associated to a development in the DCT domain similar to the one performed to generate BossBase [3], this is presented in section 2. From this analysis, we are able to decompose the set of DCT coefficients into 8 disjoint lattices where within each lattice the different coefficients are mutually uncorrelated (see section 3).

The embedding scheme is based on the conversion from the costs associated to each coefficients into an implicit zero-mean Gaussian distribution whose variance is directly computed from the costs. This "Gaussian mapping", together with the Covariance matrix estimated in section 2 enables to compute a joint Gaussian distribution and to derive its associated conditional distribution w.r.t the embedding changes performed on the previous lattices. The embedding scheme is presented in section 4. Finally section 5 presents the performance gains for different embedding strategies (UERD and J-Uniward) and different quality factors, and analyzes also the distribution of the payloads over the different lattices.

2 CORRELATIONS BETWEEN DCT COEFFICIENTS

In this section we analyze the covariance matrix between DCT coefficients of neighboring 8×8 DCT blocks after a development pipeline similar to the one used to generate BOSSBase (see Section 5.1 for more details on the development pipeline). Since the correlations related to the host content are difficult to model, we focused our analysis on the statistical model of the photonic sensor noise. We computed the covariance matrix of 3×3 neighboring blocks of size 8×8 in the DCT domain (i.e. before quantization). The covariance matrix is estimated from 1000 RAW images with constant photo-site values $\mu = 2^{12}$ coded on 14 bits and corrupted with an additive i.i.d. signal $S \sim \mathcal{N}(0, a\mu + b)$, demosaicked with the bi-linear algorithm, down-sampled to a 512×512 images, and transformed into a 2D-DCT array.

In order to take into account symmetries of the whole 576×576 covariance matrix, for example the fact that the covariance between

two horizontal neighbors is identical, the analysis of only a portion of the covariance matrix can be conducted by considering only 2×2 adjacent blocks, hence only a 256×256 covariance matrix. The scan order for the four 8×8 DCT blocks consists of a scan by rows within each block and a block-wise scan across the four blocks as shown in Figure 1.

By observing Figure 1 together with the scan order and the decomposition of the matrix into different types, we can decompose the entire covariance matrix into four different types of 64×64 matrices : one intra-block covariance matrix and three inter-block covariance matrices:

- The intra-block 64×64 covariance matrix Σ_{intra} captures the correlations between DCT coefficients of the same block.
- The horizontal and vertical covariance matrices $\Sigma_{0,1inter}$ and $\Sigma_{0,2inter}$ captures correlations between horizontal blocks and vertical blocks respectively.
- The diagonal inter-block covariance matrix $\Sigma_{0,3inter}$ captures correlation between diagonal blocks.

Important remarks can be highlighted from the analysis of these covariance matrices:

- They are sparse, i.e. lot of DCT coefficients are uncorrelated.
- Within one 8×8 DCT block, one coefficient is correlated with 6 other ones, and for vertically or horizontally adjacent blocks one coefficient is correlated with 8 other ones in each connected block. Figure 3-(b) shows for the DCT mode (0, 2) belonging to Λ_5 the locations of the correlated coefficients belonging to horizontal or vertical neighbors.
- Two diagonal blocks are nearly uncorrelated, i.e. the correlation values are very low, and in the following we consider diagonal blocks as uncorrelated.
- The patterns of the covariance matrix are immune to the type of demosaicking or down-sampling kernel. We tested the different demosaicking algorithms offered by the "rawpy" library together with different down-sampling kernels, in each case the patterns (but not the correlation values) were similar.

Note that the covariance matrix is computed in order to highlight correlations between DCT component of the sensor noise and to try to mimic them during the embedding. However, in order to be invariant to the noise power which depends of various parameters such as the sensor model or the ISO settings, we convert the *covariance* matrix into a *correlation* matrix, where each diagonal terms equals 1 and each off-diagonal term is divided by $\sigma_i \sigma_j$.

Practically, since each term of the empirical covariance matrix is defined as:

$$\Sigma_{i,j} = \frac{1}{N} \sum_{k=1}^N (C_i(k) - \bar{C}_i)(C_j(k) - \bar{C}_j), \quad (1)$$

(where \bar{C}_i is the empirical mean of coefficient C_i), each term of the correlation matrix is consequently defined as:

$$\xi_{i,j} = \frac{\Sigma_{i,j}}{\sqrt{\Sigma_{i,i}\Sigma_{j,j}}}. \quad (2)$$

Note that after this normalization, correlation coefficients which are not close to zero are rather small. For our experiments correlation coefficients for two distinct DCT modes belong to the range $[0.03; 0.07]$, but as we shall see in section 5, taking into account these correlations enables to improve the practical security of the scheme.

3 LATTICE DECOMPOSITION

From these observations we can now decompose the set of DCT coefficients into lattices where each lattice is only composed of uncorrelated coefficients. We end up with 8 lattices, because of the following observations:

- To deal with intra-block correlations, we notice that we can find 4 sets of coefficients uncorrelated to one another. The 4 subsets (lattices) $\Lambda_i \in \mathbb{N}^{16}$ with $i \in \{0, \dots, 3\}$ of these mutually decorated modes indexes are arranged thanks to a permutation matrix P such that :

$$R_{intra} = P \underbrace{\begin{bmatrix} I_{16} & \Sigma_{\Lambda_0, \Lambda_1} & \cdots & \Sigma_{\Lambda_0, \Lambda_3} \\ \Sigma_{\Lambda_1, \Lambda_0} & I_{16} & \ddots & \vdots \\ \vdots & \ddots & I_{16} & \Sigma_{\Lambda_2, \Lambda_3} \\ \Sigma_{\Lambda_3, \Lambda_0} & \cdots & \Sigma_{\Lambda_3, \Lambda_2} & I_{16} \end{bmatrix}}_{R_p^{intra}} P^{-1}$$

The displayed correlation matrix 2 after permutation of the indexes highlights the fact that within each lattice, each is only correlated with itself. However, we also notice that a coefficient belonging to Λ_i with $0 < i < 4$, is correlated with two coefficients for each other lattices belonging to the same block.

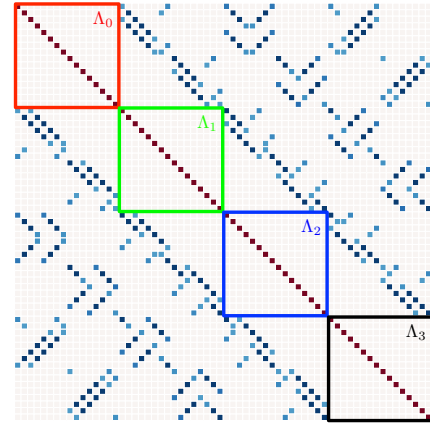


Figure 2: Intra-block correlations matrix after permutation R_p^{intra} for the 4 lattices $\{\Lambda_0, \dots, \Lambda_3\}$, colored blocks denotes the associated lattices.

- To deal with inter-block correlations, we proceed in the same way. This time, we can see from the analysis of the covariance matrix that each mode is correlated with 8 modes for each connected block (see. Figure 1, where on sub-matrices $\Sigma_{0,1inter}$ and $\Sigma_{0,2intra}$ each DCT mode is positively or negatively correlated with 8 other

coefficients). We also notice that since two diagonally-connected block are uncorrelated, we can build 2 sub-lattices of 8×8 blocks to deal inter-block correlations.

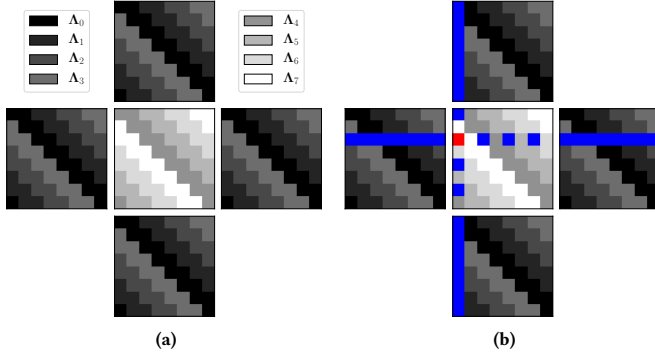


Figure 3: (a) Decomposition of the DCT modes into 8 lattices, (b) The 38 modes used to compute the conditional probability of mode $(2, 0) \in \Lambda_7$ (red), the Table 11 explicitly lists the set of correlated modes required to sample the modes from Λ_7 .

	Λ_0	Λ_1	Λ_2	Λ_3	Λ_4	Λ_5	Λ_6	Λ_7
K	0	2	4	6	32	34	36	38

Table 1: Number of correlated coefficients k for each lattice considering only previous lattices.

Based on the above considerations, each image can be split into 8 disjoint lattices in order to sample a stego signal in the DCT domain preserving both intra-block and inter-block correlations.

Figure 3 (a) shows the locations of the uncorrelated coefficients for the different lattices, and Figure 3 (b) highlights the locations of correlated coefficients belonging to previous lattices for one given mode.

Table 1 indicates for lattice Λ_1 the number of correlated coefficients, denoted K , for the lattices $\{\Lambda_{i-1}, \dots, \Lambda_0\}$. Tables 5, 6, 7, 8, 9, 10 and 11 (see the Appendices) exhibit for each mode of each lattices the different correlated modes belonging to previous lattices for the same block or adjacent ones as depicted on Figure 10.

4 EMBEDDING SCHEME

We detail now how we can leverage both the covariance matrix presented in section 2 and the lattice decomposition presented in section 3 to enable to synchronization of embedding changes for cost-based embedding schemes.

Figure 4 summarizes the different mandatory steps necessary to perform embedding which can be decomposed into five steps:

- (1) The computation of the correlation coefficients and the correlation matrix, as presented in section 2.
- (2) The decomposition of the image into 8 lattices as presented in section 3.

- (3) The computation of a covariance matrix using both the costs derived from the additive steganographic scheme and the correlation matrix. In order to do so, we convert empirical costs into Gaussian distributions. This can be justified by the fact that in order to leverage the covariance matrix of the sensor noise, we need to model the stego signal by a multivariate Gaussian distribution since it is the only distribution that can be defined only by its expectation and its covariance. The derivation of variances from costs is detailed in section 4.1.
- (4) The computation of the conditional embedding probabilities which take into account both the correlations between DCT coefficients and the modifications done on the previous lattices. This is detailed in section 4.3.
- (5) The modification of the coefficients to obtain the stego image. This is detailed in section 4.4.

4.1 From costs to Gaussian distributions

Without loss of generality, we assume that the developed steganographic scheme uses ternary embedding. For a coefficient of coordinates (i, j) into a 8×8 DCT block, we assume that the underlying unquantized stego signal is associated with a Normal distribution with zero mean and a variance $\sigma_{i,j}^2$, i.e. $S_{i,j} \sim \mathcal{N}(0, \sigma_{i,j}^2)$. As explained below, the variance is determined w.r.t both the costs computed by an heuristic algorithm (UERD or J-UNIWARD here), and to the payload size \mathbf{m} .

For each coefficient (i, j) we can compute the triplet of costs $(\rho_{i,j}^{-1}, \rho_{i,j}^0, \rho_{i,j}^{+1})$ respectively associated to the embedding changes $-1, 0, +1$. Since we use non side-informed schemes, we also assume that $\rho_{i,j}^{-1} = \rho_{i,j}^{+1}$.

We can convert the costs into embedding probabilities using Lagrangian optimization [7] by using the formula:

$$P_{i,j}(k) = \frac{\exp(-\lambda \rho_{i,j}^k)}{\exp(-\lambda \rho_{i,j}^0) + \exp(-\lambda \rho_{i,j}^{+1}) + \exp(-\lambda \rho_{i,j}^{-1})}, \quad (3)$$

with $k \in -1, 0, +1$, and λ following the payload constraint.

Denoting $q_{i,j}$ the JPEG quantization step associated to coefficient (i, j) , we now assume that the embedding probabilities correspond to the probabilities of a quantized Gaussian distribution using three quantization bins, respectively $]-\infty, -q_{i,j}/2]$, $] -q_{i,j}/2, q_{i,j}/2]$, $]q_{i,j}/2, +\infty]$ for $-1, 0, +1$. Since

$$P_{i,j}(-1) = \frac{1}{2} \operatorname{erf}\left(-\frac{q_{i,j}}{2\sqrt{2}\sigma_{i,j}}\right),$$

and $2P_{i,j}(-1) + P_{i,j}(0) = 1$, the relation between $\sigma_{i,j}^2$ and the embedding probabilities is then given by:

$$\sigma_{i,j}^2 = \frac{q_{i,j}^2}{8 (\operatorname{erf}^{-1}(P_{i,j}(0)))^2}. \quad (4)$$

4.2 Construction of the covariance matrix

The covariance matrix $\tilde{\Sigma}$ is sequentially built for each DCT coefficient of each lattice in order to take into account the embedding changes of correlated coefficient that have already been made on

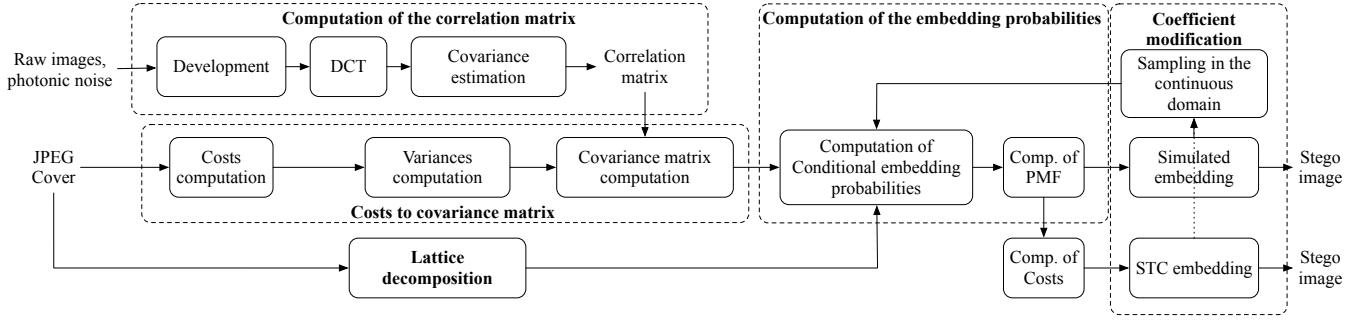


Figure 4: Overview of the embedding scheme.

the previous lattices. Its size is consequently $(K + 1) \times (K + 1)$, with K given in Table 1.

The diagonal terms of $\tilde{\Sigma}$ are given by (4) and its off-diagonal terms take into account the correlation coefficients $\xi_{i,j}$ estimated using (2).

More specifically for a given mode, the covariance matrix is built using the variances $\{\sigma_1^2, \dots, \sigma_K^2\}$ of the K correlated coefficients that have been already modified during the embedding. These variances are then weighted by the inter-correlations coefficients ξ associated to these coefficients using (2). The resulting covariance matrix $\tilde{\Sigma}$ is given by:

$$\tilde{\Sigma} = \begin{bmatrix} \sigma_1^2 & \xi_{1,2}\sigma_1\sigma_2 & \cdots & \xi_{1,m}\sigma_1\sigma_{K+1} \\ \xi_{1,2}\sigma_1\sigma_2 & \sigma_2^2 & & \\ \vdots & & \ddots & \vdots \\ \xi_{1,K+1}\sigma_1\sigma_m & \cdots & \sigma_K^2 & \sigma_{K+1}^2 \end{bmatrix}, \quad (5)$$

$$\tilde{\Sigma} \doteq \begin{bmatrix} \tilde{\Sigma}_d & \tilde{\Sigma}_c \\ \tilde{\Sigma}_r & \sigma_{K+1}^2 \end{bmatrix}, \quad (6)$$

where $\tilde{\Sigma}_d$ is the $(K \times K)$ matrix with the $(K \times K)$ first entries of $\tilde{\Sigma}$, $\tilde{\Sigma}_c$ is the $(K \times 1)$ matrix with the K first entries of the last column of $\tilde{\Sigma}$ and $\tilde{\Sigma}_r$ is the $(1 \times K)$ matrix with the K first entries of the last row of $\tilde{\Sigma}$.

4.3 Computation of embedding probabilities

We can derive the conditional pdf of $C_{K+1}|c_1, \dots, c_K$ distributed as $\mathcal{N}(\tilde{\mu}, \tilde{\sigma}^2)$, with:

$$\tilde{\mu} = \tilde{\Sigma}_r \tilde{\Sigma}_d^{-1} [c_K, \dots, c_1]^T, \quad (7)$$

$$\tilde{\sigma}^2 = \sigma_{K+1}^2 - \tilde{\Sigma}_r \tilde{\Sigma}_d^{-1} \tilde{\Sigma}_c. \quad (8)$$

Note that because of conditioning (and of synchronization), the mean of the Gaussian distribution is not anymore equal to zero. We can afterward compute the pmf by again integration over the 3 intervals $]-\infty, -q_{i,j}/2]$, $]-q_{i,j}/2, q_{i,j}/2]$, $]q_{i,j}/2, +\infty]$ for $-1, 0, +1$.

4.4 Coefficient modification

Once the pmf is computed, either we sample from it, or we convert the probabilities to costs using the relation $\rho_{i,j}^k = \log(p_{i,j}^0/p_{i,j}^k)$, and use a STC. Moreover, in order to compute (8), we need to draw

samples c_{m-1}, \dots, c_1 , which correspond to the embedding changes performed on the K DCT coefficients belonging to the previous lattices, which already are carrying a portion of the payload. This can be done for example by rejection sampling, i.e. by sampling over the Gaussian distributions until each sample belongs to the interval corresponding to the right embedding change.

5 RESULTS

5.1 Database development

In order to leverage the correlations induced by the development pipeline, we explain in this section the development pipeline used to develop the raw images of BOSSBase. Since this database is composed of images coming from different cameras, the sensors have different sizes (from CR2 of size 2602×3906 , to DNG of size 3472×5216 , NEF of size 2014×3039 , and PEF files of size 3124×4688), thus to be able to have the same down-sampling factor for each image it is important to find the minimum length or width dimension for all the images. As a result, for each image we developed the image using bi-linear demosaicking, luminance averaging, bilinear downscaling, and then performed a centered crop of width and height equal to $l_{min} = 2014$ before performing JPEG compression to build our BOSSBase-SD (Same Dimensions). Note that except for the crop operation and the demosaicking and down-sampling kernels, this database is very similar to the BOSSBase database.

5.2 Benchmark setup

The empirical security is evaluated as the minimal total classification error probability under equal priors, $P_E = \min_{P_{FA}} \frac{1}{2} (P_{FA} + P_{MD})$, with P_{FA} and P_{MD} standing for the false-alarm and missed detection rates. The JPEG images are steganalyzed with the DCTR feature set [9] and the low-complexity linear classifier [4], training and testing sets both equal to 2×5000 , the P_E is averaged after 10 random splits on training and testing sets.

The presented adaptations, named *Cov-J-UNIWARD* and *Cov-UERD* (which use respectively the costs computed by J-UNIWARD and UERD) are compared with *J-UNIWARD* and *UERD*. However, since the synchronized version of these algorithms use conditioning, the achievable entropy is slightly attenuated of about 1% as can be seen on Table 2. Consequently, in order to make a fair comparison we have compared J-UNIWARD and UERD to their synchronized versions by using the payload size computed from *Cov-J-UNIWARD* and *Cov-UERD* respectively to J-UNIWARD and UERD (see also

Figure 5). This operation is performed over the whole image base for H_{in} (bits/nzAC) $\in \{0.1, 0.2, \dots, 1.0\}$.

Note also that the embedding is performed by considering quantization steps of size 1, which correspond to a targeted JPEG quality factor of 100. Extensive test with appropriate values of q are left for future researches.

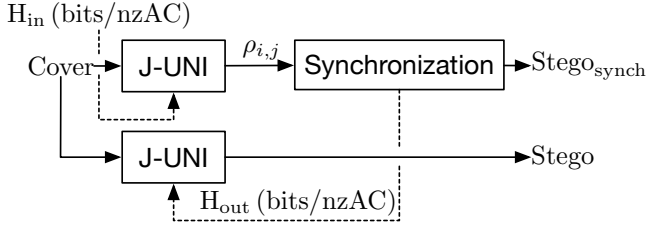


Figure 5: Embedding setup to ensure that the stego image carry the payload in the case of a J-UNIWARD embedding.

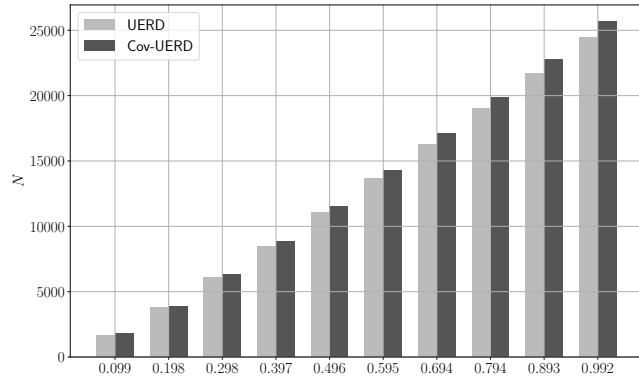


Figure 6: Comparison between the number of modifications for UERD and Cov-UERD for same embedding rates.

5.3 Comparison with UERD and J-UNIWARD

Results for both schemes are presented in Table 4 for an embedding rate of 0.28 bpnzac and for a range of embedding rates within $[0, 1.0]$ on Figures 8 and 7.

Several observations can be made:

- The JPEG quality factor offering the best performance improvement over the classical schemes is $QF95$ for both schemes.
- This improvement can be substantial with a maximum gain of around 7% for both schemes at 0.28 bpnzac. On the other hand for high embedding rates (i.e. ≥ 0.4 bpnzac), the impact of synchronization can be either negative for J-Uniward, or nonexistent for UERD. This can be due to the fact that the model of the stego signal for these additive schemes might be too different with the model of the stego signal of the sensor noise, which makes the synchronization either useless or detrimental.
- The costs provided by UERD seem on average to be more suited to the synchronization procedure than J-Uniward, which at $QF75$ for example, does not show any gain.

5.4 Effects of synchronization

The synchronization w.r.t. previous embedding changes on previous lattices naturally induces fluctuations in the final embedding probabilities. One can observe on figure 9 that if the same embedding changes are performed on DCT coefficients belonging to lattice Λ_0 between the synchronized and the non-synchronized version of UERD, the embedding probabilities on other coefficients belonging to $\{\Lambda_1, \dots, \Lambda_7\}$ can undergo important bias, going up to ± 0.15 for several coefficients.

Table 3 presents the average entropy for one sample image for each lattice at $QF95$. One can notice a small decrease of the entropy between Λ_k and Λ_{k+4} ($k \in \{0, \dots, 3\}$) which corresponds to same DCT modes on two adjacent diagonal blocks, and which is due to synchronization. This behavior can be explained by the fact that for two random coefficients (C_1, C_2) coding the same DCT mode belonging to two vertically or horizontally connected blocks, $H(C_2|C_1) \leq H(C_2)$.

Figure 6 compare the number of embedding changes between UERD and Cov-UERD for same embedding rates and one sample image. Logically the proposed synchronization procedure induces more embedding changes, but at the same time decreases the detectability for small embedding rates.

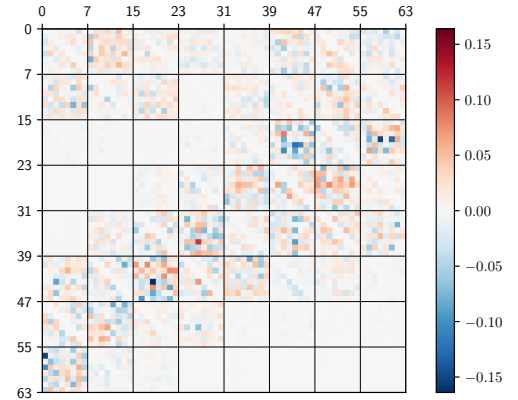


Figure 9: Difference of probabilities map to sample a +1 between UERD and Cov-UERD for a sample image (cropped to a 64×64 array), $QF100$, 0.48 bpnzac. Identical embedding changes for the two schemes have been performed on coefficients belonging to lattice Λ_0 .

5.5 Complexity

This embedding algorithm is computationally expensive because the complexity of computing the conditional distribution increases with the complexity of the Cholesky decomposition of the covariance matrix, i.e., as $O(n^3)$ where $n = K + 1$, which depend of which lattice the considered mode belongs : $n = 1$ for $m \in \Lambda_0$, $n = 3$ for $m \in \Lambda_3$ and $n = 39$ for $m \in \Lambda_7$. On a 1.6 GHz Intel Core i5, our python implementation of simulated embedding on a 512×512 image is performed in 1min 46s while an UERD simulated embedding takes 2 seconds.

Targeted	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
True	0.099	0.198	0.298	0.397	0.496	0.595	0.694	0.794	0.893	0.992

Table 2: Targeted payload vs True embedding rate in pbznac due to synchronization for one sample image.

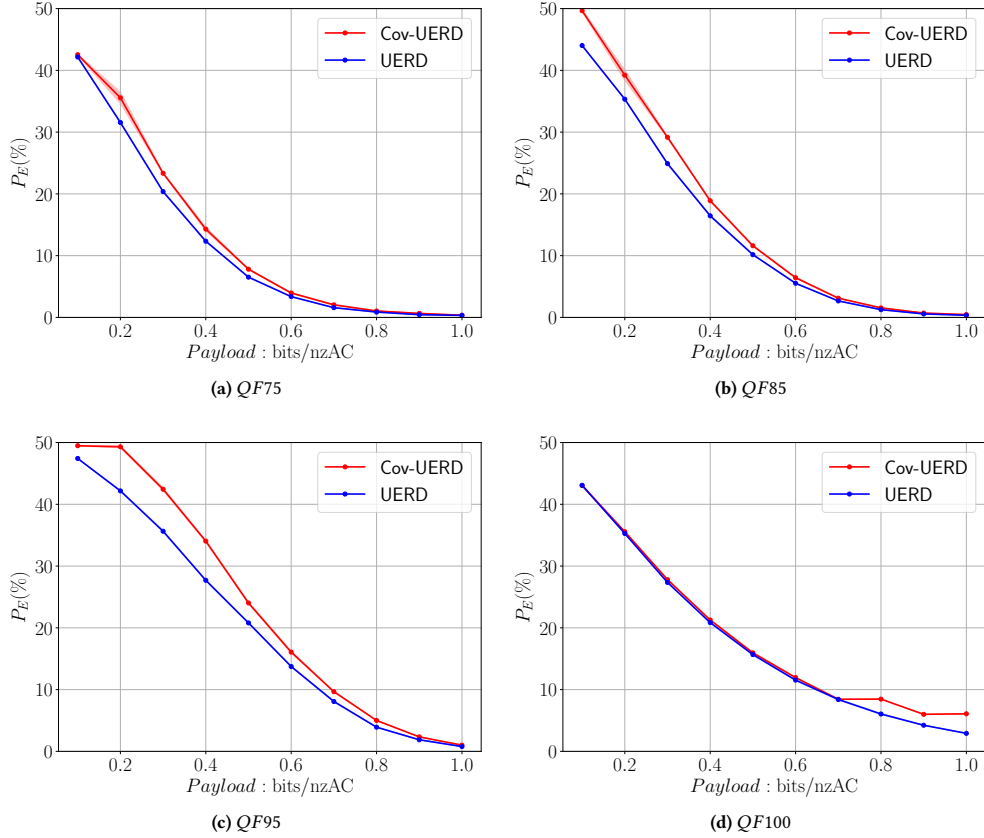


Figure 7: UERD and its synchronized version $QF \in \{75, 95, 85, 100\}$ for respectively (a), (b), (c) and (d).

Λ_0	Λ_1	Λ_2	Λ_3	Λ_4	Λ_5	Λ_6	Λ_7
1.94	0.95	0.92	1.52	1.82	0.88	0.86	1.43

Table 3: Average entropy by coefficients ($\times 10^{-2}$) over the 8 lattices for QF95, for a targetted payload of 0.3 bpnzAC on one sample image.

6 CONCLUSIONS AND PERSPECTIVES

We have proposed a synchronization mechanism for JPEG steganography that can be used for classical additive cost-based embedding schemes. The synchronization is done by leveraging the correlations between DCT coefficients after the development from RAW to DCT of an image composed of photonic noise. The embedding scheme requires the use of 8 lattices of disjoint DCT coefficients in order to synchronize one coefficient with potentially 6 coefficients of the same block and 24 coefficients belonging to adjacent horizontal or diagonal blocks. The correlations are taken into account

P_E (%) / JPEG QF	Cov-UERD	UERD	Cov-JUNI	JUNI
75	23.341 \pm 0.116	20.368 \pm 0.08	21.089 \pm 0.104	21.606 \pm 0.059
85	29.167 \pm 0.145	24.896 \pm 0.11	27.62 \pm 0.136	27.269 \pm 0.109
95	42.442 \pm 0.242	35.64 \pm 0.11	45.282 \pm 0.113	37.205 \pm 0.045
100	27.797 \pm 0.094	27.351 \pm 0.069	33.129 \pm 0.08	31.733 \pm 0.095

Table 4: Average empirical security (P_E in %) and associated standard deviation over 10 runs for different quality factors and embedding strategies on BOSSBase SD with bilinear demosaicking, and downscaling but the same payload of 0.28 bpnzAC. DCTR features combined with regularized linear classifier are used for steganalysis.

by converting classical heuristic costs into marginal Gaussian distributions, and then building a multivariate Gaussian distribution associated with a covariance matrix that takes into the correlation between DCT coefficients.

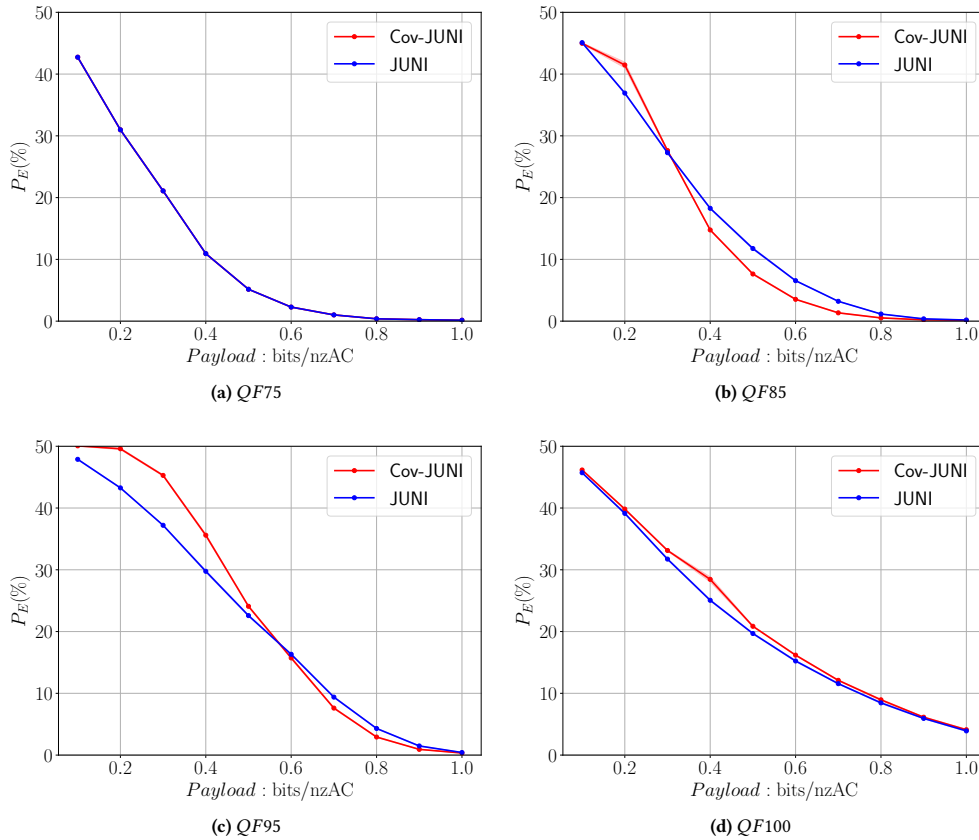


Figure 8: J-UNIWARD and its synchronized version for $QF \in \{75, 95, 85, 100\}$ for respectively (a), (b), (c) and (d).

Our encouraging results show that this methods enables to increase the practical security by around 7% for an embedding rate of 0.28 bpnzac at QF95.

7 APPENDICES

The appendices present, for each DCT mode of each lattice, the list of correlated modes belonging to previous lattices.

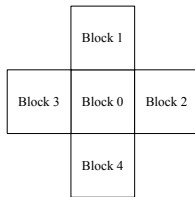


Figure 10: Block naming convention.

8 ACKNOWLEDGMENTS

This work has been funded in part by the French National Research Agency (ANR-18-ASTR-0009), ALASKA project: <https://alaska.utt.fr>, and by the French ANR DEFALS program (ANR-16-DEFA-0003).

REFERENCES

- [1] Patrick Bas. Steganography via Cover-Source Switching. 2016. IEEE Workshop on Information Forensics and Security (WIFS).
- [2] Patrick Bas. An embedding mechanism for Natural Steganography after down-sampling. 2017. IEEE ICASSP.
- [3] Patrick Bas, Tomas Filler, and Tomas Pevny. "Break Our Steganographic System": The Ins and Outs of Organizing BOSS. In *INFORMATION HIDING*, volume 6958/2011 of *Lecture Notes in Computer Science*, pages 59–70, Czech Republic, September 2011.
- [4] Rémi Cogranne, Vahid Sedighi, Jessica Fridrich, and Tomáš Pevný. Is ensemble classifier needed for steganalysis in high-dimensional feature spaces? In *Information Forensics and Security (WIFS), 2015 IEEE International Workshop on*, pages 1–6. IEEE, 2015.
- [5] Tomáš Denemark and Jessica Fridrich. Improving steganographic security by synchronizing the selection channel. In *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*, pages 5–14. ACM, 2015.
- [6] Tomáš Filler and Jessica Fridrich. Gibbs construction in steganography. *IEEE Transactions on Information Forensics and Security*, 5(4):705–720, 2010.
- [7] Tomas Filler, Jan Judas, and Jessica Fridrich. Minimizing additive distortion in steganography using syndrome-trellis codes. *Information Forensics and Security, IEEE Transactions on*, 6(3):920–935, 2011.
- [8] Linjie Guo, Jiangqun Ni, Wenkang Su, Chengpei Tang, and Yun-Qing Shi. Using statistical image model for jpeg steganography: Uniform embedding revisited. *IEEE Transactions on Information Forensics and Security*, 10(12):2669–2680, 2015.
- [9] Vojtěch Holub and Jessica Fridrich. Low-complexity features for jpeg steganalysis using undecimated dct. *IEEE Transactions on Information Forensics and Security*, 10(2):219–228, 2015.
- [10] Vojtěch Holub, Jessica Fridrich, and Tomáš Denemark. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security*, 2014(1):1–13, 2014.
- [11] Bin Li, Ming Wang, Xiaolong Li, Shunquan Tan, and Jiwu Huang. A strategy of clustering modification directions in spatial image steganography. *Information*

Mode / Block	(0, 2)	(1, 3)	(2, 4)	(3, 5)	(4, 6)	(5, 7)	(7, 1)	(6, 0)	(0, 3)	(1, 4)	(2, 5)	(3, 6)	(4, 7)	(7, 2)	(6, 1)	(5, 0)
1	(0, 2)	(1, 3)	(2, 4)	(3, 5)	(4, 6)	(5, 7)	(7, 1)	(6, 0)	(0, 3)	(1, 4)	(2, 5)	(3, 6)	(4, 7)	(7, 2)	(6, 1)	(5, 0)
	(0, 0)	(3, 3)	(4, 4)	(5, 5)	(6, 6)	(7, 7)	(7, 7)	(0, 0)	(0, 1)	(1, 2)	(4, 5)	(5, 6)	(6, 7)	(7, 0)	(0, 1)	(5, 6)
	(2, 2)	(1, 1)	(2, 2)	(3, 3)	(4, 4)	(5, 5)	(1, 1)	(6, 6)	(2, 3)	(3, 4)	(2, 3)	(3, 4)	(4, 5)	(1, 2)	(6, 7)	(7, 0)

Table 5: Correlated modes for each mode of Λ_1 w.r.t. coefficients belonging to the previous lattice.

Mode / Block	(0, 4)	(1, 5)	(2, 6)	(3, 7)	(7, 3)	(6, 2)	(5, 1)	(4, 0)	(0, 5)	(1, 6)	(2, 7)	(7, 4)	(6, 3)	(5, 2)	(4, 1)	(3, 0)
1	(0, 4)	(1, 5)	(2, 6)	(3, 7)	(7, 3)	(6, 2)	(5, 1)	(4, 0)	(0, 5)	(1, 6)	(2, 7)	(7, 4)	(6, 3)	(5, 2)	(4, 1)	(3, 0)
	(0, 2)	(1, 3)	(4, 6)	(5, 7)	(7, 5)	(6, 4)	(3, 1)	(2, 0)	(0, 3)	(5, 6)	(4, 7)	(7, 2)	(6, 5)	(3, 2)	(2, 1)	(5, 0)
	(0, 6)	(5, 5)	(6, 6)	(1, 7)	(7, 1)	(4, 2)	(1, 1)	(6, 0)	(0, 1)	(3, 6)	(6, 7)	(7, 6)	(6, 1)	(5, 4)	(4, 5)	(1, 0)
	(0, 0)	(1, 1)	(2, 4)	(7, 7)	(7, 7)	(2, 2)	(5, 5)	(4, 4)	(4, 5)	(1, 4)	(2, 3)	(7, 0)	(4, 3)	(1, 2)	(6, 1)	(3, 4)
	(4, 4)	(3, 5)	(2, 2)	(3, 5)	(1, 3)	(6, 6)	(5, 3)	(0, 0)	(6, 5)	(1, 2)	(2, 1)	(1, 4)	(2, 3)	(5, 6)	(4, 3)	(3, 6)
	(2, 4)	(7, 5)	(2, 0)	(3, 3)	(5, 3)	(0, 2)	(5, 7)	(4, 6)	(2, 5)	(1, 0)	(0, 7)	(5, 4)	(0, 3)	(7, 2)	(0, 1)	(7, 0)

Table 6: Correlated modes for each mode of Λ_2 w.r.t. coefficients belonging to the previous lattices.

Mode / Block	(0, 6)	(1, 7)	(7, 5)	(6, 4)	(5, 3)	(4, 2)	(3, 1)	(2, 0)	(0, 7)	(7, 6)	(6, 5)	(5, 4)	(4, 3)	(3, 2)	(2, 1)	(1, 0)
1	(0, 6)	(1, 7)	(7, 5)	(6, 4)	(5, 3)	(4, 2)	(3, 1)	(2, 0)	(0, 7)	(7, 6)	(6, 5)	(5, 4)	(4, 3)	(3, 2)	(2, 1)	(1, 0)
	(0, 4)	(5, 7)	(7, 3)	(6, 2)	(3, 3)	(4, 4)	(5, 1)	(4, 0)	(4, 7)	(7, 2)	(6, 3)	(3, 4)	(4, 5)	(3, 4)	(4, 1)	(5, 0)
	(0, 2)	(3, 7)	(7, 1)	(6, 6)	(5, 5)	(2, 2)	(3, 5)	(6, 0)	(6, 7)	(7, 4)	(6, 1)	(5, 2)	(2, 3)	(5, 2)	(2, 5)	(3, 0)
	(4, 6)	(1, 3)	(7, 7)	(4, 4)	(1, 3)	(6, 2)	(1, 1)	(2, 4)	(0, 1)	(7, 0)	(4, 5)	(1, 4)	(6, 3)	(3, 6)	(6, 1)	(1, 4)
	(6, 6)	(7, 7)	(1, 5)	(2, 4)	(5, 1)	(4, 6)	(3, 3)	(2, 6)	(0, 3)	(3, 6)	(2, 5)	(5, 6)	(4, 1)	(1, 2)	(2, 3)	(1, 6)
	(0, 0)	(1, 1)	(3, 5)	(6, 0)	(5, 7)	(0, 2)	(7, 1)	(0, 0)	(2, 7)	(1, 6)	(0, 5)	(5, 0)	(0, 3)	(7, 2)	(2, 7)	(1, 2)
	(2, 6)	(1, 5)	(5, 5)	(0, 4)	(7, 3)	(4, 0)	(3, 7)	(2, 2)	(0, 5)	(5, 6)	(6, 7)	(7, 4)	(4, 7)	(3, 0)	(0, 1)	(7, 0)

Table 7: Correlated modes for each mode of Λ_3 w.r.t. coefficients belonging to the previous lattices.

Mode / Block	(0, 0)	(1, 1)	(2, 2)	(3, 3)	(4, 4)	(5, 5)	(6, 6)	(7, 7)	(0, 1)	(1, 2)	(2, 3)	(3, 4)	(4, 5)	(5, 6)	(6, 7)	(7, 0)	
1	(0, 0)	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)	(0, 6)	(0, 7)	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)	(0, 6)	(0, 7)	(0, 0)	
	(1, 0)	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(1, 6)	(1, 7)	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(1, 6)	(1, 7)	(1, 0)	
	(2, 0)	(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)	(2, 6)	(2, 7)	(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)	(2, 6)	(2, 7)	(2, 0)	
	(3, 0)	(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)	(3, 6)	(3, 7)	(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)	(3, 6)	(3, 7)	(3, 0)	
	(4, 0)	(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)	(4, 6)	(4, 7)	(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)	(4, 6)	(4, 7)	(4, 0)	
	(5, 0)	(5, 1)	(5, 2)	(5, 3)	(5, 4)	(5, 5)	(5, 6)	(5, 7)	(5, 1)	(5, 2)	(5, 3)	(5, 4)	(5, 5)	(5, 6)	(5, 7)	(5, 0)	
	(6, 0)	(6, 1)	(6, 2)	(6, 3)	(6, 4)	(6, 5)	(6, 6)	(6, 7)	(6, 1)	(6, 2)	(6, 3)	(6, 4)	(6, 5)	(6, 6)	(6, 7)	(6, 0)	
	(7, 0)	(7, 1)	(7, 2)	(7, 3)	(7, 4)	(7, 5)	(7, 6)	(7, 7)	(7, 1)	(7, 2)	(7, 3)	(7, 4)	(7, 5)	(7, 6)	(7, 7)	(7, 0)	
	2	(0, 0)	(1, 0)	(2, 0)	(3, 0)	(4, 0)	(5, 0)	(6, 0)	(7, 0)	(0, 0)	(1, 0)	(2, 0)	(3, 0)	(4, 0)	(5, 0)	(6, 0)	(7, 0)
		(0, 1)	(1, 1)	(2, 1)	(3, 1)	(4, 1)	(5, 1)	(6, 1)	(7, 1)	(0, 1)	(1, 1)	(2, 1)	(3, 1)	(4, 1)	(5, 1)	(6, 1)	(7, 1)
		(0, 2)	(1, 2)	(2, 2)	(3, 2)	(4, 2)	(5, 2)	(6, 2)	(7, 2)	(0, 2)	(1, 2)	(2, 2)	(3, 2)	(4, 2)	(5, 2)	(6, 2)	(7, 2)
		(0, 3)	(1, 3)	(2, 3)	(3, 3)	(4, 3)	(5, 3)	(6, 3)	(7, 3)	(0, 3)	(1, 3)	(2, 3)	(3, 3)	(4, 3)	(5, 3)	(6, 3)	(7, 3)
		(0, 4)	(1, 4)	(2, 4)	(3, 4)	(4, 4)	(5, 4)	(6, 4)	(7, 4)	(0, 4)	(1, 4)	(2, 4)	(3, 4)	(4, 4)	(5, 4)	(6, 4)	(7, 4)
		(0, 5)	(1, 5)	(2, 5)	(3, 5)	(4, 5)	(5, 5)	(6, 5)	(7, 5)	(0, 5)	(1, 5)	(2, 5)	(3, 5)	(4, 5)	(5, 5)	(6, 5)	(7, 5)
		(0, 6)	(1, 6)	(2, 6)	(3, 6)	(4, 6)	(5, 6)	(6, 6)	(7, 6)	(0, 6)	(1, 6)	(2, 6)	(3, 6)	(4, 6)	(5, 6)	(6, 6)	(7, 6)
		(0, 7)	(1, 7)	(2, 7)	(3, 7)	(4, 7)	(5, 7)	(6, 7)	(7, 7)	(0, 7)	(1, 7)	(2, 7)	(3, 7)	(4, 7)	(5, 7)	(6, 7)	(7, 7)
3		(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)	(0, 0)
		(0, 1)	(1, 1)	(2, 1)	(3, 1)	(4, 1)	(5, 1)	(6, 1)	(7, 1)	(0, 1)	(1, 1)	(2, 1)	(3, 1)	(4, 1)	(5, 1)	(6, 1)	(7, 1)
		(0, 2)	(1, 2)	(2, 2)	(3, 2)	(4, 2)	(5, 2)	(6, 2)	(7, 2)	(0, 2)	(1, 2)	(2, 2)	(3, 2)	(4, 2)	(5, 2)	(6, 2)	(7, 2)
		(0, 3)	(1, 3)	(2, 3)	(3, 3)	(4, 3)	(5, 3)	(6, 3)	(7, 3)	(0, 3)	(1, 3)	(2, 3)	(3, 3)	(4, 3)	(5, 3)	(6, 3)	(7, 3)
		(0, 4)	(1, 4)	(2, 4)	(3, 4)	(4, 4)	(5, 4)	(6, 4)	(7, 4)	(0, 4)	(1, 4)	(2, 4)	(3, 4)	(4, 4)	(5, 4)	(6, 4)	(7, 4)
		(0, 5)	(1, 5)	(2, 5)	(3, 5)	(4, 5)	(5, 5)	(6, 5)	(7, 5)	(0, 5)	(1, 5)	(2, 5)	(3, 5)	(4, 5)	(5, 5)	(6, 5)	(7, 5)
		(0, 6)	(1, 6)	(2, 6)	(3, 6)	(4, 6)	(5, 6)	(6, 6)	(7, 6)	(0, 6)	(1, 6)	(2, 6)	(3, 6)	(4, 6)	(5, 6)	(6, 6)	(7, 6)
		(0, 7)	(1, 7)	(2, 7)	(3, 7)	(4, 7)	(5, 7)	(6, 7)	(7, 7)	(0, 7)	(1, 7)	(2, 7)	(3, 7)	(4, 7)	(5, 7)	(6, 7)	(7, 7)
	4	(0, 0)	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)	(0, 6)	(0, 7)	(0, 1)	(0, 2)	(0, 3)	(0, 4)	(0, 5)	(0, 6)	(0, 7)	(0, 0)
		(1, 0)	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(1, 6)	(1, 7)	(1, 1)	(1, 2)	(1, 3)	(1, 4)	(1, 5)	(1, 6)	(1, 7)	(1, 0)
		(2, 0)	(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)	(2, 6)	(2, 7)	(2, 1)	(2, 2)	(2, 3)	(2, 4)	(2, 5)	(2, 6)	(2, 7)	(2, 0)
		(3, 0)	(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)	(3, 6)	(3, 7)	(3, 1)	(3, 2)	(3, 3)	(3, 4)	(3, 5)	(3, 6)	(3, 7)	(3, 0)
		(4, 0)	(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)	(4, 6)	(4, 7)	(4, 1)	(4, 2)	(4, 3)	(4, 4)	(4, 5)	(4, 6)	(4, 7)	(4, 0)
		(5, 0)	(5, 1)	(5, 2)	(5, 3)	(5, 4)	(5, 5)	(5, 6)	(5, 7)	(5, 1)	(5, 2)	(5, 3)	(5, 4)	(5, 5)	(5, 6)	(5, 7)	(5, 0)
		(6, 0)	(6, 1)	(6, 2)	(6, 3)	(6, 4)	(6, 5)	(6, 6)	(6, 7)	(6, 1)	(6, 2)	(6, 3)	(6, 4)	(6, 5)	(6, 6)	(6, 7)	(6, 0)
		(7, 0)	(7, 1)	(7, 2)	(7, 3)	(7, 4)	(7, 5)	(7, 6)	(7, 7)	(7, 1)	(7, 2)	(7, 3)	(7, 4)	(7, 5)	(7, 6)	(7, 7)	(7, 0)
0		(0, 0)	(1, 1)	(2, 1)	(3, 3)	(4, 4)	(5, 5)	(6, 6)	(7, 7)	(0, 1)	(1, 2)	(2, 3)	(3, 4)	(4, 5)	(5, 6)	(6, 7)	(7, 0)

Table 8: Correlated modes for each mode of Λ_4 w.r.t. coefficients belonging to the previous lattices.

[12] Weixiang Li, Weiming Zhang, Kejiang Chen, Wenbo Zhou, and Nenghai Yu. Defining joint distortion for jpeg steganography. In *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*, pages 5–16. ACM, 2018.

[13] Taburet Théo, Bas Patrick, Sawaya Wadid, and Jessica Fridrich. Natural steganography in jpeg domain with a linear development pipeline, 2020.

[14] Weiming Zhang, Zhuo Zhang, Lili Zhang, Hanyi Li, and Nenghai Yu. Decomposing joint distortion for adaptive steganography. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(10):2274–2280, 2016.

Mode / Block	(0,6)	(1,7)	(7,5)	(6,4)	(5,3)	(4,2)	(3,1)	(2,0)	(0,7)	(7,6)	(6,5)	(5,4)	(4,3)	(3,2)	(2,1)	(1,0)
1	(0,6)	(0,7)	(0,5)	(0,4)	(0,3)	(0,2)	(0,1)	(0,0)	(0,7)	(0,6)	(0,5)	(0,4)	(0,3)	(0,2)	(0,1)	(0,0)
	(1,6)	(1,7)	(1,5)	(1,4)	(1,3)	(1,2)	(1,1)	(1,0)	(1,7)	(1,6)	(1,5)	(1,4)	(1,3)	(1,2)	(1,1)	(1,0)
	(2,6)	(2,7)	(2,5)	(2,4)	(2,3)	(2,2)	(2,1)	(2,0)	(2,7)	(2,6)	(2,5)	(2,4)	(2,3)	(2,2)	(2,1)	(2,0)
	(3,6)	(3,7)	(3,5)	(3,4)	(3,3)	(3,2)	(3,1)	(3,0)	(3,7)	(3,6)	(3,5)	(3,4)	(3,3)	(3,2)	(3,1)	(3,0)
	(4,6)	(4,7)	(4,5)	(4,4)	(4,3)	(4,2)	(4,1)	(4,0)	(4,7)	(4,6)	(4,5)	(4,4)	(4,3)	(4,2)	(4,1)	(4,0)
	(5,6)	(5,7)	(5,5)	(5,4)	(5,3)	(5,2)	(5,1)	(5,0)	(5,7)	(5,6)	(5,5)	(5,4)	(5,3)	(5,2)	(5,1)	(5,0)
	(6,6)	(6,7)	(6,5)	(6,4)	(6,3)	(6,2)	(6,1)	(6,0)	(6,7)	(6,6)	(6,5)	(6,4)	(6,3)	(6,2)	(6,1)	(6,0)
	(7,6)	(7,7)	(7,5)	(7,4)	(7,3)	(7,2)	(7,1)	(7,0)	(7,7)	(7,6)	(7,5)	(7,4)	(7,3)	(7,2)	(7,1)	(7,0)
	(0,0)	(1,0)	(7,0)	(6,0)	(5,0)	(4,0)	(3,0)	(2,0)	(0,0)	(7,0)	(6,0)	(5,0)	(4,0)	(3,0)	(2,0)	(1,0)
2	(0,1)	(1,1)	(7,1)	(6,1)	(5,1)	(4,1)	(3,1)	(2,1)	(0,1)	(7,1)	(6,1)	(5,1)	(4,1)	(3,1)	(2,1)	(1,1)
	(0,2)	(1,2)	(7,2)	(6,2)	(5,2)	(4,2)	(3,2)	(2,2)	(0,2)	(7,2)	(6,2)	(5,2)	(4,2)	(3,2)	(2,2)	(1,2)
	(0,3)	(1,3)	(7,3)	(6,3)	(5,3)	(4,3)	(3,3)	(2,3)	(0,3)	(7,3)	(6,3)	(5,3)	(4,3)	(3,3)	(2,3)	(1,3)
	(0,4)	(1,4)	(7,4)	(6,4)	(5,4)	(4,4)	(3,4)	(2,4)	(0,4)	(7,4)	(6,4)	(5,4)	(4,4)	(3,4)	(2,4)	(1,4)
	(0,5)	(1,5)	(7,5)	(6,5)	(5,5)	(4,5)	(3,5)	(2,5)	(0,5)	(7,5)	(6,5)	(5,5)	(4,5)	(3,5)	(2,5)	(1,5)
	(0,6)	(1,6)	(7,6)	(6,6)	(5,6)	(4,6)	(3,6)	(2,6)	(0,6)	(7,6)	(6,6)	(5,6)	(4,6)	(3,6)	(2,6)	(1,6)
	(0,7)	(1,7)	(7,7)	(6,7)	(5,7)	(4,7)	(3,7)	(2,7)	(0,7)	(7,7)	(6,7)	(5,7)	(4,7)	(3,7)	(2,7)	(1,7)
	(0,0)	(1,0)	(7,0)	(6,0)	(5,0)	(4,0)	(3,0)	(2,0)	(0,0)	(7,0)	(6,0)	(5,0)	(4,0)	(3,0)	(2,0)	(1,0)
	(0,1)	(1,1)	(7,1)	(6,1)	(5,1)	(4,1)	(3,1)	(2,1)	(0,1)	(7,1)	(6,1)	(5,1)	(4,1)	(3,1)	(2,1)	(1,1)
3	(0,2)	(1,2)	(7,2)	(6,2)	(5,2)	(4,2)	(3,2)	(2,2)	(0,2)	(7,2)	(6,2)	(5,2)	(4,2)	(3,2)	(2,2)	(1,2)
	(0,3)	(1,3)	(7,3)	(6,3)	(5,3)	(4,3)	(3,3)	(2,3)	(0,3)	(7,3)	(6,3)	(5,3)	(4,3)	(3,3)	(2,3)	(1,3)
	(0,4)	(1,4)	(7,4)	(6,4)	(5,4)	(4,4)	(3,4)	(2,4)	(0,4)	(7,4)	(6,4)	(5,4)	(4,4)	(3,4)	(2,4)	(1,4)
	(0,5)	(1,5)	(7,5)	(6,5)	(5,5)	(4,5)	(3,5)	(2,5)	(0,5)	(7,5)	(6,5)	(5,5)	(4,5)	(3,5)	(2,5)	(1,5)
	(0,6)	(1,6)	(7,6)	(6,6)	(5,6)	(4,6)	(3,6)	(2,6)	(0,6)	(7,6)	(6,6)	(5,6)	(4,6)	(3,6)	(2,6)	(1,6)
	(0,7)	(1,7)	(7,7)	(6,7)	(5,7)	(4,7)	(3,7)	(2,7)	(0,7)	(7,7)	(6,7)	(5,7)	(4,7)	(3,7)	(2,7)	(1,7)
	(0,0)	(1,0)	(7,0)	(6,0)	(5,0)	(4,0)	(3,0)	(2,0)	(0,0)	(7,0)	(6,0)	(5,0)	(4,0)	(3,0)	(2,0)	(1,0)
	(0,1)	(1,1)	(7,1)	(6,1)	(5,1)	(4,1)	(3,1)	(2,1)	(0,1)	(7,1)	(6,1)	(5,1)	(4,1)	(3,1)	(2,1)	(1,1)
	(0,2)	(1,2)	(7,2)	(6,2)	(5,2)	(4,2)	(3,2)	(2,2)	(0,2)	(7,2)	(6,2)	(5,2)	(4,2)	(3,2)	(2,2)	(1,2)
4	(0,3)	(1,3)	(7,3)	(6,3)	(5,3)	(4,3)	(3,3)	(2,3)	(0,3)	(7,3)	(6,3)	(5,3)	(4,3)	(3,3)	(2,3)	(1,3)
	(0,4)	(1,4)	(7,4)	(6,4)	(5,4)	(4,4)	(3,4)	(2,4)	(0,4)	(7,4)	(6,4)	(5,4)	(4,4)	(3,4)	(2,4)	(1,4)
	(0,5)	(1,5)	(7,5)	(6,5)	(5,5)	(4,5)	(3,5)	(2,5)	(0,5)	(7,5)	(6,5)	(5,5)	(4,5)	(3,5)	(2,5)	(1,5)
	(0,6)	(1,6)	(7,6)	(6,6)	(5,6)	(4,6)	(3,6)	(2,6)	(0,6)	(7,6)	(6,6)	(5,6)	(4,6)	(3,6)	(2,6)	(1,6)
	(0,7)	(1,7)	(7,7)	(6,7)	(5,7)	(4,7)	(3,7)	(2,7)	(0,7)	(7,7)	(6,7)	(5,7)	(4,7)	(3,7)	(2,7)	(1,7)
	(0,0)	(1,0)	(7,0)	(6,0)	(5,0)	(4,0)	(3,0)	(2,0)	(0,0)	(7,0)	(6,0)	(5,0)	(4,0)	(3,0)	(2,0)	(1,0)
	(1,6)	(1,7)	(1,5)	(1,4)	(1,3)	(1,2)	(1,1)	(1,0)	(1,7)	(1,6)	(1,5)	(1,4)	(1,3)	(1,2)	(1,1)	(1,0)
	(2,6)	(2,7)	(2,5)	(2,4)	(2,3)	(2,2)	(2,1)	(2,0)	(2,7)	(2,6)	(2,5)	(2,4)	(2,3)	(2,2)	(2,1)	(2,0)
	(3,6)	(3,7)	(3,5)	(3,4)	(3,3)	(3,2)	(3,1)	(3,0)	(3,7)	(3,6)	(3,5)	(3,4)	(3,3)	(3,2)	(3,1)	(3,0)
0	(4,6)	(4,7)	(4,5)	(4,4)	(4,3)	(4,2)	(4,1)	(4,0)	(4,7)	(4,6)	(4,5)	(4,4)	(4,3)	(4,2)	(4,1)	(4,0)
	(5,6)	(5,7)	(5,5)	(5,4)	(5,3)	(5,2)	(5,1)	(5,0)	(5,7)	(5,6)	(5,5)	(5,4)	(5,3)	(5,2)	(5,1)	(5,0)
	(6,6)	(6,7)	(6,5)	(6,4)	(6,3)	(6,2)	(6,1)	(6,0)	(6,7)	(6,6)	(6,5)	(6,4)	(6,3)	(6,2)	(6,1)	(6,0)
	(7,6)	(7,7)	(7,5)	(7,4)	(7,3)	(7,2)	(7,1)	(7,0)	(7,7)	(7,6)	(7,5)	(7,4)	(7,3)	(7,2)	(7,1)	(7,0)
	(0,6)	(1,7)	(7,5)	(6,4)	(5,3)	(4,2)	(3,1)	(2,0)	(0,7)	(7,6)	(6,5)	(5,4)	(4,3)	(3,2)	(2,1)	(1,0)
	(0,4)	(5,7)	(7,3)	(6,2)	(3,3)	(4,4)	(5,1)	(4,0)	(4,7)	(7,2)	(6,3)	(3,4)	(4,5)	(3,4)	(4,1)	(5,0)
	(0,2)	(3,7)	(7,1)	(6,6)	(5,5)	(2,2)	(3,5)	(6,0)	(6,7)	(7,4)	(6,1)	(5,2)	(2,3)	(5,2)	(2,5)	(3,0)
	(4,6)	(1,3)	(7,7)	(4,4)	(1,3)	(6,2)	(1,1)	(2,4)	(0,1)	(7,0)	(4,5)	(1,4)	(6,3)	(3,6)	(6,1)	(1,4)
	(6,6)	(7,7)	(1,5)	(2,4)	(5,1)	(4,6)	(3,3)	(2,6)	(0,3)	(3,6)	(2,5)	(5,6)	(4,1)	(1,2)	(2,3)	(1,6)
(0,0)	(1,1)	(3,5)	(6,0)	(5,7)	(0,2)	(7,1)	(0,0)	(2,7)	(1,6)	(0,5)	(5,0)	(0,3)	(7,2)	(2,7)	(1,2)	
(2,6)	(1,5)	(5,5)	(0,4)	(7,3)	(4,0)	(3,7)	(2,2)	(0,5)	(5,6)	(6,7)	(7,4)	(4,7)	(3,0)	(0,1)	(7,0)	

Table 11: Correlated modes for each mode of Λ_7 w.r.t. coefficients belonging to the previous lattices. The 38 modes (depicted in Figure 3) used to compute the conditional probability of mode $(2, 0) \in \Lambda_7$ are colored in blue while and the red one correspond to itself.