

TARGETED POOLED LATENT-SPACE STEGANALYSIS APPLIED TO GENERATIVE STEGANOGRAPHY, WITH A FIX

Etienne Levecque³, Aurelien Noirault¹, Tomas Pevny², Jan Butora¹, Patrick Bas¹, Rémi Cogranne³

1. UMR 9189 CRISAL
Univ. Lille, CNRS,
Centrale Lille, Lille, France

2. Artificial Intelligence Center
Czech Technical University
Prague, Czech Republic.

3. LIST3N, UTT
University of Techn. of Troyes,
Troyes, France

ABSTRACT

Steganographic schemes dedicated to generated images modify the seed vector in the latent space to embed a message. Whereas most steganalysis methods attempt to detect the embedding in the image space, this paper proposes to perform steganalysis in the latent space by modeling the statistical distribution of the norm of the latent vector. Specifically, we analyze the practical security of a scheme proposed by Hu *et al.* [5] for latent diffusion models, which is both robust and practically undetectable when steganalysis is performed on generated images. We show that after embedding, the Stego (latent) vector is distributed on a hypersphere while the Cover vector is i.i.d. Gaussian. By going from the image space to the latent space, we show that it is possible to model the norm of the vector in the latent space under the Cover or Stego hypothesis as Gaussian distributions with different variances. A Likelihood Ratio Test is then derived to perform pooled steganalysis. The impact of the potential knowledge of the prompt and the number of diffusion steps is also studied. Additionally, we show how, by randomly sampling the norm of the latent vector before generation, the initial Stego scheme becomes undetectable in the latent space.

Index Terms— Steganalysis, Generative Steganography, Latent space, statistical tests

1. INTRODUCTION

Steganography on generated images is a recent active field since it enables the embedding of large payloads and is rather robust to operations such as JPEG compression, while being possibly undetectable. One notable feature of generative steganography is the fact that, contrary to classical steganography [4], which only alters the noise component of the image, the embedding also alters its semantic content. Generative steganography methods [5,8,12] are popular among latent diffusion models [6] (LDM) since these models provide high-quality images. In this case, the embedding strategy consists of partitioning the latent space (typically of size [4, 64, 64])

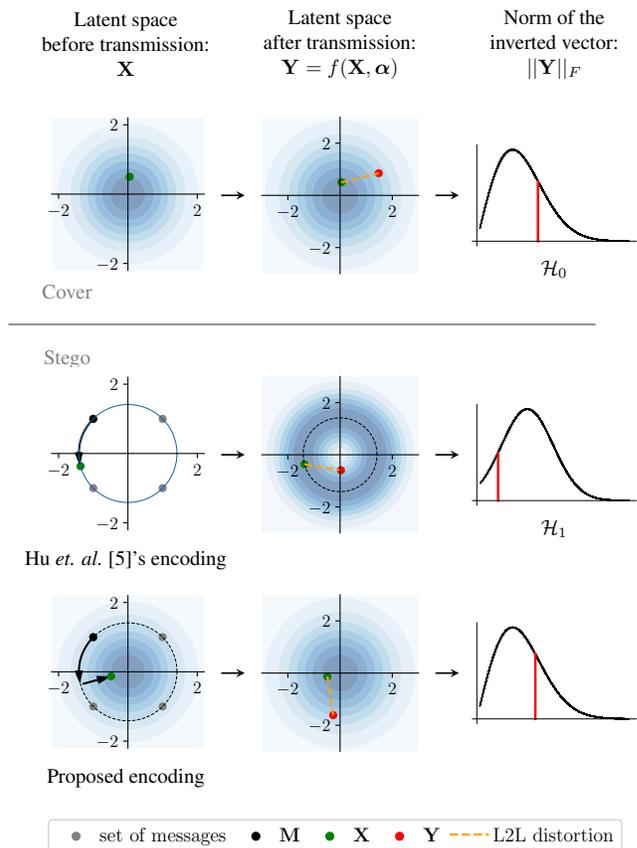


Fig. 1. Illustration in 2D: gray dots in the Stego setup represent the space of messages using the encoding proposed in [5] (2 bits are encoded), which creates a spherical distribution in the original latent space and can be approximated as a Gaussian distribution after the latent-to-latent channel. Our proposal, consisting of sampling the norm after encoding, fixes this discrepancy.

into key-dependent and message-dependent regions and moving the seed into such a region before generation. Unfortunately, the whole latent-to-latent (L2L) process (consisting of generating, coding, and going back to the latent space) can

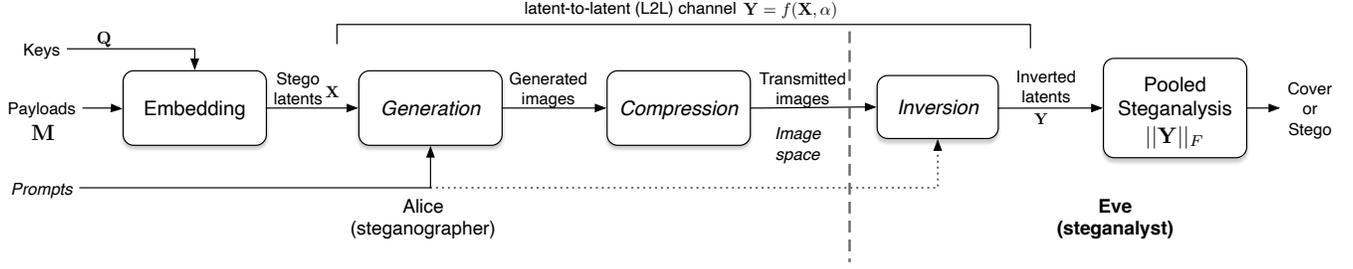


Fig. 2. Principle of the presented steganalysis scheme targeted to [5]. The entities in *italic* are, according to the Kerckhoffs principle, considered as public. Prompts are potentially public (dotted arrow).

be considered as a noisy process and may produce decoding errors. Consequently, one way to preserve robustness is to modify the distribution of the initially i.i.d. Gaussian distribution in a secret key-dependent subspace of the latent space in order to create clusters around the codewords related to each embedded message.

If this alteration may, in principle, increase the detectability of the Stego scheme, many published schemes are blind to security analysis in the latent space. The first class of steganalysis schemes assesses the detectability by training classical steganalysis networks [1, 9] to distinguish between generated Cover¹ images and generated Stego images. The second class of security analysis proposes to compute the Kullback-Leibler Divergence on marginals in the latent space [10, 11] in order to show that the embedding has no impact on the distribution.

Our contribution focuses on the generative steganographic scheme proposed by Hu *et. al.* which is both robust to the inversion channel (the Bit Error Rate is on average smaller than 2%) and not detectable by steganalysis in the image domain. The originality of our detection scheme relies on the fact that the steganalysis is neither performed in the image space nor on marginal distributions in the latent space, but on the joint distribution in the latent space. Moreover, since the detection is based on a Likelihood Ratio Test (LRT), it is easy to derive a pooled detector based on a batch of images.

2. BINARY MAPPING IN THE LATENT SPACE

2.1. Notations and security hypothesis

Let \mathbf{M} be a secret message of n bits cast to ± 1 and reshaped to match the latent space dimension $(4, \sqrt{n}/2, \sqrt{n}/2)$. In this paper, the latent space is of dimension $(4, 64, 64)$ and thus $n = 16384$. The latent-to-latent (L2L) channel is composed of three parts always in the same order: a generation process that maps the seed to an image, an operation inducing distortion such as compression, and the inverse generation that maps an image back to a vector \mathbf{Y} in the latent space. This channel is noted f such as $\mathbf{Y} = f(\mathbf{X}, \alpha)$ with α a vector

¹for the sake of simplicity and consistency with more than 20 years of steganography, we decide to keep naming non-Stego images as "Covers".

of hyperparameters such as the prompt, the guidance, or the compression.

In this steganography setup, Alice and Bob have to agree on a secret key \mathbf{k} and on a set of parameters α which will be fixed for a given message. Note, however, that according to Kerckhoffs' principle, Eve, the steganalyst, *has access* to the L2L channel f and its fixed parameters α but *does not have access* to the secret key.

2.2. Description of the embedding scheme

This scheme, presented by Hu *et. al.* [5], is particularly suited for LDM since it is robust to the L2L channel. It relies on the principle of Spread Spectrum watermarking [3] (SS) where each embedded bit is spread on a secret carrier. To embed a secret message, Alice (the steganographer) performs the modulation using an orthonormal key-dependent pseudo-random matrix \mathbf{Q} (it is the result of a Gram-Schmidt decomposition) using the following multiplication:

$$\mathbf{X} = \mathbf{Q} \cdot \mathbf{M} \cdot \mathbf{Q}^T, \quad (1)$$

Note that the matrix representation of the message \mathbf{M} enables \mathbf{Q} to have a separable transform where each carrier associated to the binary modulation $m_{i,j} \in \{-1, +1\}$ equals $\mathbf{C}_{i,j} = \mathbf{q}_i \cdot \mathbf{q}_j^T$, with \mathbf{q}_k the k^{th} column of \mathbf{Q} .

This latent seed \mathbf{X} is sent through the L2L channel f :

$$\mathbf{Y} = f(\mathbf{X}, \alpha). \quad (2)$$

Bob, the receiver, can then estimate the payload $\hat{\mathbf{M}}$ by projecting on each carrier with the following operation:

$$\hat{\mathbf{M}} = \mathbf{Q}^T \cdot \mathbf{Y} \cdot \mathbf{Q}. \quad (3)$$

2.3. Model of the norm before L2L

Let $R_{\mathbf{X}}$ be the random variable associated with the norm of the latent vector before the L2L channel. We recall that, under the null hypothesis (image is Cover) $\mathbf{X} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_n)$ with \mathbf{I}_n the identity matrix. The norm of a Gaussian distribution is a Chi distribution with n degrees of freedom and thus still under

the null hypothesis, $R_{\mathbf{X}} \sim \chi_n$. Finally, since the dimension of the latent vector n is large, we can use the following limit:

$$\lim_{n \rightarrow +\infty} R_{\mathbf{X}} \xrightarrow{d} \mathcal{N}\left(\sqrt{n}, \frac{1}{2}\right). \quad (4)$$

Under the alternative hypothesis (image is Stego), $\mathbf{X} = \mathbf{Q} \cdot \mathbf{M} \cdot \mathbf{Q}^T$ and thus we have that $R_{\mathbf{X}} = \|\mathbf{M}\|_F$ because the norm is invariant by rotation. Moreover, we know that \mathbf{M} is distributed on the vertices of the n -dimensional hypercube with edge length 2. Therefore, the distance between any vertex and the origin is always half the diagonal of the hypercube, which equals \sqrt{n} . Under the Stego hypothesis, the norm of the latent vector before the L2L channel is constant and is equal to \sqrt{n} and $R_{\mathbf{X}} \sim \delta_{\sqrt{n}}$ with δ the Dirac distribution.

$$\begin{cases} \mathcal{H}_0 : R_{\mathbf{X}} \sim \mathcal{N}\left(\sqrt{n}, \frac{1}{2}\right), \\ \mathcal{H}_1 : R_{\mathbf{X}} \sim \delta_{\sqrt{n}}. \end{cases} \quad (5)$$

2.4. Model of the norm after L2L

We define $R_{\mathbf{Y}}$ as the random variable associated with the norm of the latent vector after the L2L channel. We recall that $\mathbf{Y} = f(\mathbf{X}, \alpha)$. To model the norm $R_{\mathbf{Y}}$ of the latent vector \mathbf{Y} , we assume that the L2L channel induces a distortion $E(\alpha)$ on the norm of \mathbf{X} and that this distortion follows a Gaussian distribution: $\varepsilon(\alpha) \sim \mathcal{N}(0, \sigma^2(\alpha))$ and thus, $R_{\mathbf{Y}} = R_{\mathbf{X}} + E(\alpha)$.

$$\begin{cases} \mathcal{H}_0 : R_{\mathbf{Y}} \sim \mathcal{N}\left(\sqrt{n}, \frac{1}{2} + \sigma^2(\alpha)\right), \\ \mathcal{H}_1 : R_{\mathbf{Y}} \sim \mathcal{N}\left(\sqrt{n}, \sigma^2(\alpha)\right). \end{cases} \quad (6)$$

We can see that both hypotheses only differ in the variance, and this difference should vanish if the L2L is very noisy. On the contrary, if the L2L is more and more accurate, the difference of variances will become maximal.

2.5. Scaled SS to fix distribution discrepancy

With classical SS encoding, we have seen that, under the Stego hypothesis, \mathbf{X} is uniformly distributed on the hypersphere of radius \sqrt{n} . However, under the Cover hypothesis, \mathbf{X} should follow a Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I}_n)$ and its norm should follow a χ_n -distribution. So, the solution to transform the uniform hyper-sphere distribution into a standard Gaussian one is to sample a norm s according to the χ_n -distribution and then to scale the norm of the latent vector as follows:

$$\mathbf{X} = \frac{s}{\sqrt{n}} \mathbf{Q} \cdot \mathbf{M} \cdot \mathbf{Q}^T. \quad (7)$$

With this fix, we can verify that the norm of \mathbf{X} is equal to s and thus follows a χ_n -distribution:

$$\|\mathbf{X}\|_F = \frac{s}{\sqrt{n}} \times \|\mathbf{M}\|_F = s. \quad (8)$$

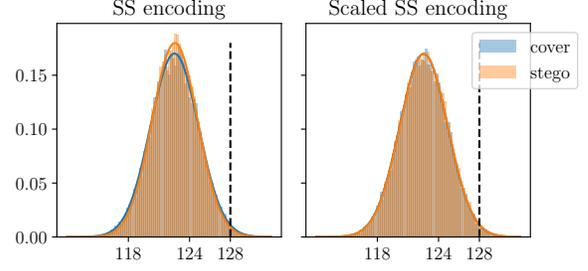


Fig. 3. Histogram of the norm of the latent vector $\|\mathbf{Y}\|_F$. The curves represent the Gaussian fit for each class. Notice the small difference in the variance on the left plot, which is corrected on the right plot. Guidance=5.0, steps=20, 20k prompts.

To recover the payload $\hat{\mathbf{M}}$ from the latent vector \mathbf{Y} , Bob needs to apply the inverse rotation and observes the sign of each coefficient:

$$\hat{\mathbf{M}} = \text{sign}(\mathbf{Q}^T \cdot \mathbf{Y} \cdot \mathbf{Q}). \quad (9)$$

Fig. 3 illustrates how the fix acts on the norm of the latent vector $\|\mathbf{Y}\|_F$ received by Bob. Without the scaling, the Cover and the Stego distribution do not have the same variance as the norm, but with the scaling, this difference vanishes. Note that the mean is 122.5 for all 4 distributions instead of 128. With SS encoding, the estimated variance is 5.50 for the Covers and 4.91 for the Stegos, with a difference of 0.61. These small differences with the statistical model can be explained by the assumption on f .

3. STATISTICAL STEGANALYSIS

3.1. Likelihood Ratio Test

The proposed attack is based on a likelihood ratio test (LRT) on the norm $R_{\mathbf{Y}}$ of the latent vector. As shown previously, this norm follows a Gaussian distribution whose parameters depend on the hypothesis. In theory, the variance is the main source of discrepancy, while the mean should be identical. However, our theoretical model relies on assumptions about f , and as shown in Fig. 3, the mean is not exactly $\sqrt{n} = 128$ and the variance $\sigma^2(\alpha)$ is unknown. To address this, we estimate the mean and variance using Maximum Likelihood Estimation. Since the mean is assumed to be identical under both hypotheses, we denote by $\hat{\mu}$ the estimated mean of both the Cover and Stego distributions, and by $\hat{\sigma}_0^2$ (respectively $\hat{\sigma}_1^2$) the estimated variance of the Cover (respectively Stego) distribution. Let $r_{\mathbf{Y}}$ be a realization of $R_{\mathbf{Y}}$; the LRT for a single image is then given by:

$$\Lambda(r_{\mathbf{Y}}) = \frac{\mathcal{N}(r_{\mathbf{Y}}; \hat{\mu}, \hat{\sigma}_0^2)}{\mathcal{N}(r_{\mathbf{Y}}; \hat{\mu}, \hat{\sigma}_1^2)}. \quad (10)$$

The null hypothesis is rejected if $\Lambda(r_{\mathbf{Y}}) < \tau$, where τ is a decision threshold, and the image is classified as Stego; otherwise, it is classified as Cover. The choice of τ is discussed in Section 3.2.

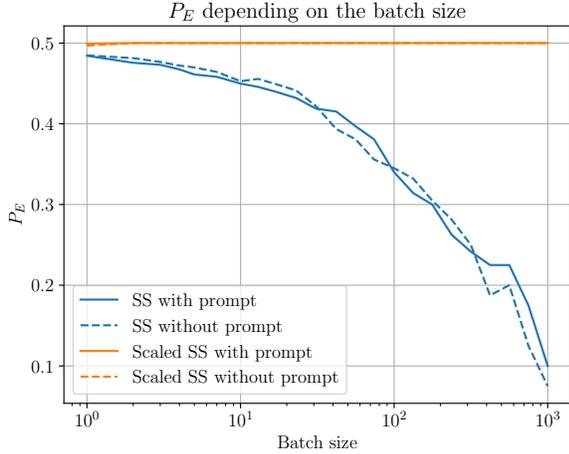


Fig. 4. P_E for steps = 20: as the batch size increases, the LRT becomes more powerful, contrary to the proposed fix, which is effective.

This LRT can be extended to a batch of images to increase its power. Assuming that a batch of N images belongs to a single class (Cover or Stego), and denoting by $B = (r_{\mathbf{Y}_i})_i$ the batch of norms, the independence assumption yields the following LRT:

$$\Lambda(B) = \prod_{i=1}^N \frac{\mathcal{N}(r_{\mathbf{Y}_i}; \hat{\mu}, \hat{\sigma}_0^2)}{\mathcal{N}(r_{\mathbf{Y}_i}; \hat{\mu}, \hat{\sigma}_1^2)}. \quad (11)$$

3.2. Experimental results

All the experiments have been done with StableDiffusion 1.5 because it was the version used by Hu *et al.*, but the attack should work with any diffusion model. In their paper, they used a guidance of 5.0 for generation and inversion and 20 diffusion steps, so we fixed these parameters to the same values. The prompts are randomly selected from the DiffusionDB [7] database. To avoid any prompt mismatch, each randomly selected prompt was used to generate both a Cover image with a random seed \mathbf{X} and a Stego image with a payload as the seed. A total of 20k prompts were used in each experiment, resulting in 20k Stego images and 20k Cover images.

To evaluate our test, we used the Probability of Error (P_E) defined as:

$$P_E = \min_{P_{FA}} \frac{P_{FA} + P_{MD}(P_{FA})}{2}, \quad (12)$$

where P_{FA} is the probability of false alarm and P_{MD} is the probability of missed detection.

Concerning the decision threshold τ , the estimation of the mean and variance becomes noisy for larger batch sizes because fewer samples are available. This noisiness creates false alarms, so we scale the threshold based on the batch size. The expression of the threshold τ is:

$$\tau = \exp(-\log(N)/20), \quad (13)$$

P_E (%)	Steps	
	20	50
Batch size	20	50
1	48.4	46.4
10	45.0	40.9
100	34.0	30.0

Table 1. P_E in percentage for different numbers of diffusion steps.

BER (%)	With prompt	Without prompt
SS [5]	97.53	97.56
Scaled SS	97.53	97.52

Table 2. Impact of the knowledge of the prompt. Guidance=5.0, steps=20.

with N the batch size. If the estimation is not noisy, it is possible to keep $\tau = 1$ for any batch size. Finally, to evaluate the robustness of the Stego encoding, we use the Bit Error Rate (BER), defined as the proportion of correct bits decoded.

For both experiments, we used a 20-fold cross-validation and averaged the P_E across each fold. The estimated mean and variance of the norm are computed on 1k Cover and 1k Stego images, while the remaining 19k Cover and 19k Stego images are used to compute P_E . Fig. 4 shows the probability of error of the attack, which starts around 48% for a single image and decreases with increasing batch size, while the proposed scaled SS encoding achieves a near-perfect 50% P_E . Table 1 reports P_E values for different numbers of diffusion steps for both generation and inversion in the L2L channel, showing that more diffusion steps reduce channel distortion and improve attack effectiveness. Moreover, as shown in Table 2, the proposed scaled encoding does not alter the BER of the original encoding. Finally, Fig. 4 suggests that even without knowledge of the prompt or the L2L parameters α , Eve can still perform the attack with the same effectiveness.

4. CONCLUSIONS AND PERSPECTIVES

This paper shows that if the message embedding for generative steganography is not detectable in the image space or on marginal distributions in the latent space, the inversion process makes the steganalysis still possible when relying on multivariate distributions, for example, by deriving a test on the norm of the vector, like in this contribution. We also show that mimicking the multivariate distribution in the latent space, here by sampling the norm of the seed, is a way to remove this weakness and to achieve Stego-security [2], which means that the distributions of Cover and Stego vectors are identical.

However, since the inversion channel is noisy, the detection process depends on inversion parameters, such as the number of steps or prior knowledge of the prompt. Given that the quality of the generative process heavily relies on the denoising processes, attacks in latent space will likely become more popular in the future.

5. ACKNOWLEDGMENTS

This work received funding from the French Defense & Innovation Agency. This work was also supported by a French government grant managed by the Agence Nationale de la Recherche under the France 2030 program, reference ANR-22-PECY-0011. This work was also supported by a grant managed by the Grant Agency of the Czech Republic, reference GACR 25-17259K.

6. REFERENCES

- [1] Mehdi Boroumand, Mo Chen, and Jessica Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5):1181–1193, 2018.
- [2] Francois Cayre and Patrick Bas. Kerckhoffs-based embedding security classes for woa data hiding. *IEEE Transactions on Information Forensics and Security*, 3(1):1–15, 2008.
- [3] Ingemar J Cox, Matthew L Miller, Jeffrey A Bloom, Jessica Fridrich, and Ton Kalker. Digital watermarking. *Morgan Kaufmann Publishers*, 54:56–59, 2008.
- [4] Jessica Fridrich. *Steganography in digital media: principles, algorithms, and applications*. Cambridge university press, 2009.
- [5] Xiaoxiao Hu, Sheng Li, Qichao Ying, Wanli Peng, Xinpeng Zhang, and Zhenxing Qian. Establishing robust generative image steganography via popular stable diffusion. *IEEE Transactions on Information Forensics and Security*, 2024.
- [6] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.
- [7] Zijie J. Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. DiffusionDB: A large-scale prompt gallery dataset for text-to-image generative models. *arXiv:2210.14896 [cs]*, 2022.
- [8] Zijin Yang, Kai Zeng, Kejiang Chen, Han Fang, Weiming Zhang, and Nenghai Yu. Gaussian shading: Provable performance-lossless image watermarking for diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12162–12171, 2024.
- [9] M. Yedroudj, F. Comby, and M. Chaumont. Yedroudjnet: An efficient CNN for spatial steganalysis. In *IEEE ICASSP*, pages 2092–2096, Alberta, Canada, April 15–20, 2018.
- [10] Ru Zhang, Shiqi Dong, and Jianyi Liu. Invisible steganography via generative adversarial networks. *Multimedia tools and applications*, 78(7):8559–8575, 2019.
- [11] Zhuo Zhang, Jia Liu, Yan Ke, Yu Lei, Jun Li, Minqing Zhang, and Xiaoyuan Yang. Generative steganography by sampling. *IEEE access*, 7:118586–118597, 2019.
- [12] Qing Zhou, Ping Wei, Zhenxing Qian, Xinpeng Zhang, and Sheng Li. Improved generative steganography based on diffusion model. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.