TOWARDS A CORRECT USAGE OF CRYPTOGRAPHY IN SEMANTIC WATERMARKS FOR DIFFUSION MODELS

Jonas Thietke, Andreas Müller , Denis Lukovnikov, Asja Fischer*, Erwin Quiring*
Ruhr University Bochum, Germany
jonas.thietke@rub.de

ABSTRACT

Semantic watermarking methods enable the direct integration of watermarks into the generation process of latent diffusion models by only modifying the initial latent noise. One line of approaches building on Gaussian Shading relies on cryptographic primitives to steer the sampling process of the latent noise. However, we identify several issues in the usage of cryptographic techniques in Gaussian Shading, particularly in its proof of lossless performance and key management, causing ambiguity in follow-up works, too. In this work, we therefore revisit the cryptographic primitives for semantic watermarking. We introduce a novel, general proof of lossless performance based on IND$-CPA security for semantic watermarks. We then discuss the configuration of the cryptographic primitives in semantic watermarks with respect to security, efficiency, and generation quality.

1 INTRODUCTION

Inversion-based semantic watermarks are a novel class of watermarking methods for latent diffusion models (LDMs) (Wen et al., 2023; Yang et al., 2024; Ci et al., 2024; Gunn et al., 2024). These watermarks change the initial latent noise to contain a watermark pattern which is recovered later by inverting the denoising process in a diffusion model. Hence, the diffusion model does not need to be changed. Semantic watermarks are thus easy to implement and empirical results suggest a high robustness to image perturbations. As only the initial latent is changed, the watermark becomes a plausible, inherent part of the generated image, for instance, through specific object details.

The semantic watermarking methods differ in how they modify the initial latent noise z_T and can be categorized into two types. *Distribution-changing* methods such as Tree-Ring (Wen et al., 2023) and RingID (Ci et al., 2024) add fixed circular patterns into the frequency spectrum of z_T which changes the distribution of z_T. In contrast, *distribution-preserving* methods such as Gaussian Shading (Yang et al., 2024) and PRC Gunn et al. (2024) keep the distribution of z_T. They critically rely on *cryptographic primitives* to generate a pseudorandom sequence that steers the sampling process of z_T.

However, Gaussian Shading, laying the foundation for distribution-preserving semantic watermarking, has not properly specified the cryptographic primitives. This can cause considerable ambiguity on the usage of this watermark method. In particular, we find that the proof of lossless performance, proving the watermark's undetectability, is not accurately specified. Moreover, it only covers the scenario where each user generates only a single image ever. The implications for key management are not discussed either, leading to multiple possible configurations of Gaussian Shading.

In this work, we revisit the cryptographic primitives for semantic watermarking. We focus on Gaussian Shading, but also discuss PRC. First, we present a novel, general proof based on IND$-CPA security (Rogaway, 2004) that allows demonstrating the lossless performance of a semantic watermark. It also covers the realistic watermarking scenario where a user generates multiple images. We apply this proof on Gaussian Shading and discuss its applicability to PRC.

Second, we analyze the implications of this proof for the key management in semantic watermarking regarding security, efficiency, and generation quality and variety. We show that the secure configuration of Gaussian Shading does not affect the generation quality and variety, but leads to a rather inefficient scheme in terms of runtime and storage. PRC, in contrast, can also be deployed efficiently.

*Equal supervision

2 BACKGROUND

In this section, we shortly describe the applied cryptographic principles and the watermarking process of Gaussian Shading. Without loss of generality, we focus on the multi-bit scenario where the watermark allows detection and user identification.

Stream Ciphers. One of the key advances in Gaussian Shading is the use of a stream cipher to guide the image generation process. In general, a stream cipher is an encryption algorithm that encrypts a message s by generating a keystream which is combined with the message to create a ciphertext that can be securely transmitted. In a nutshell, the cipher works as follows. First, a pseudo-random generator (PRNG) is used to obtain a keystream, $K = \mathrm{PRNG}(k, \eta)$, where k is a secret key and η is a public nonce. Afterwards, the message is encrypted using bitwise XOR: $c = K \oplus s$. Note that c looks like a random bit string, which is an essential property for its use in Gaussian Shading. In order to decrypt c, the receiver first uses PRNG to obtain the same keystream K as used for encryption, and then obtains the original message by using bitwise XOR: $s = K \oplus c$. In summary, the encryption function is $\mathrm{Encr}(k, \eta, s) = \mathrm{PRNG}(k, \eta) \oplus s$, and the decryption function is $\mathrm{Decr}(k, \eta, c) = \mathrm{PRNG}(k, \eta) \oplus c$. We refer to Katz & Lindell (2015) for a more elaborate explanation. A secure stream cipher should ensure that every change of even one bit in k and η entirely changes the output of PRNG. Otherwise, various attacks would be possible (Boura & Naya-Plasencia, 2023). Gaussian Shading applies ChaCha20 (Bernstein et al., 2008) as stream cipher, which requires a 256-bit secret key k and a 96-bit nonce η.

Watermark Generation And Verification. Figure 2 in Appendix A illustrates the watermarking process of Gaussian Shading (Yang et al., 2024). Before generating images, the provider generates a random user id m for each user. When the user prompts the provider to generate a new image given some textual description, the provider first samples a latent z_T with a sampling strategy $\mathcal{S}$ and subsequently uses it to generate a new image x using the generator $\mathcal{G}$ through iterative denoising starting from z_T. Gaussian Shading is integrated in the first step by changing the default sampling strategy based on a standard Gaussian $\mathcal{N}(O, I)$. Gaussian Shading instead uses the user identifier m to steer the sampling of z_T. First, m is replicated several times to increase robustness during message recovery, yielding $s = \mathrm{Repl}(m)$. The replicated user id s is then encrypted using a stream cipher: $c = \mathrm{Encr}(k, \eta, s)$. This encrypted message c is used to steer the sampling procedure $\mathcal{S}$. To this end, the standard normal distribution is divided into 2^ℓ bins, each with equal probability mass. The elements of c are used to select the bins of $\mathcal{N}(0, 1)$ from which each element of the random vector z_T is sampled. For $\ell = 1$, we have two bins and randomly sample either a negative or a positive value $z_T[i]$ from the Gaussian distribution depending on the binary value of $c[i]$. In summary, we can describe the process to obtain a watermarked image x as a concatenation of multiple functions: $x = \mathrm{GS}(k, \eta, m) = \mathcal{G}(\mathcal{S}(\mathrm{Encr}(k,\ \eta,\ \mathrm{Repl}(m))))$.

For watermark verification of an image x', the model provider performs a full inversion $\mathcal{I}$ to get an estimated latent noise $\hat{z}_T = \mathcal{I}(x')$. Next, the inverse sampling process $\mathcal{S}^{-1}$ is done where $\hat{z}_T$ is quantized to obtain the encrypted message bits $\hat{c}$. After decrypting $\hat{c}$ with $\mathrm{Decr}(k, \eta, \hat{c})$ and applying error correction with Repl^{-1}, we obtain the recovered user id $\hat{m}$. Taken together, this process can be described as follows: $\hat{m} = \mathrm{GS}^{-1}(k, \eta, x') = \mathrm{Repl}^{-1}(\mathrm{Decr}(k,\ \eta,\ \mathcal{S}^{-1}(\mathcal{I}(x'))))$. The final stage is to check if $\hat{m}$ matches with any user id m known by the service provider. This is done by comparing the number of matched bits between $\hat{m}$ and every known m. A match is found if the number of matching bits exceeds a pre-defined threshold.

3 REVISITING CRYPTOGRAPHY FOR SEMANTIC WATERMARKING

We proceed with a critical assessment of the cryptographic principles used in Gaussian Shading. As described earlier, a symmetric stream cipher is used to create a pseudorandom message that controls the sampling process. This has several advantages. The distribution of c becomes uniform, so that it can be shown that the sampling process still results in a Gaussian distribution (Yang et al., 2024). Moreover, it enables proving undetectability. Yang et al. show that the watermark created by Gaussian Shading does not introduce any pattern in the image by providing *a proof of lossless performance*. It builds on a security definition from steganography and states the following: If there is no polynomial time algorithm that can tell if an image is watermarked without having the secret

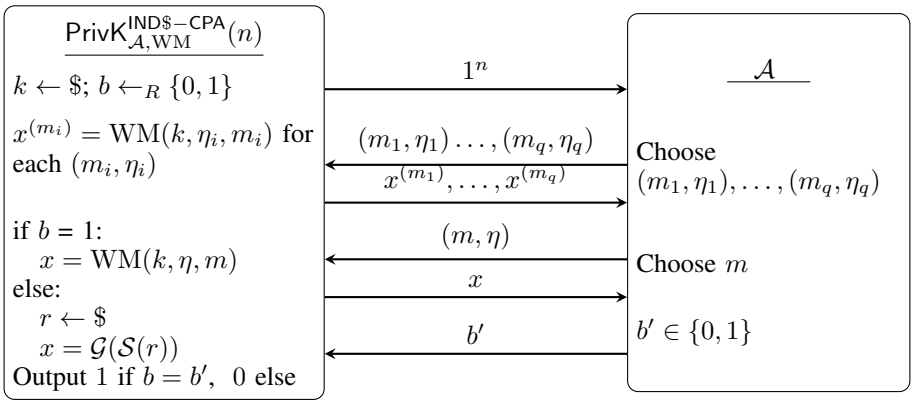

Figure 1: IND$-CPA game for a watermarking scheme WM

key, then there cannot be any patterns in the image. This means it is not possible to detect the watermark by a simple pattern recognition algorithm.

However, we identify several issues. The definition made by Gaussian Shading is an unclear specification of Hopper et al. (2002). Neither does it cover the actual usage of watermarking with multiple images that are generated and need to be watermarked. Furthermore, the key management is not specified, causing a considerable ambiguity regarding the deployment of Gaussian Shading, which has already affected further work building on Gaussian Shading (see Section 3.2).

In the following, we address these cryptographic shortcomings of Gaussian Shading with the aim to provide general insights on how cryptography should be used for semantic watermarking. In particular, we first propose to use IND$-CPA security Rogaway (2004) as a more rigorous security definition and show how to establish the undetectability of semantic watermarks within this framework. We apply this proof to Gaussian Shading and outline its applicability to PRC. Second, we examine the implications for the key management and discuss how a semantic watermark needs to be deployed to fulfill the undetectability proof. We discuss both Gaussian Shading and PRC.

3.1 A Novel Proof of Lossless Performance

In cryptography, a standard security assumption is IND-CPA (Indistinguishability under Chosen Plaintext Attack). Informally, this means that an adversary $\mathcal{A}$ cannot determine which message was encrypted, even if $\mathcal{A}$ knows the two possible messages and has observed encryptions of other (not necessarily different) messages before (Katz & Lindell, 2015). IND$-CPA is a slightly stronger assumption, stating that $\mathcal{A}$ cannot distinguish the encryption of a known message from a random bit string without knowing the key. Formally, we define **the game** $\mathsf{PrivK}^{\mathsf{IND\$-CPA}}_{\mathcal{A},\mathrm{WM}}(n)$, which is a polynomial time algorithm and depicted in Figure 1. It uses the watermarking algorithm $\mathrm{WM}(k, \eta, m)$ to output a watermarked image using the secret key and a nonce which introduces randomness. In the beginning, it generates a random secret key k and transmits the security parameter as 1^n to $\mathcal{A}$.

In the **first phase** of the game, the adversary can request up to q watermarked images for messages and nonces (m_i, η_i) that $\mathcal{A}$ provides. The inputs can be identical or different. The game responds with the corresponding watermarked images $x^{(m_i)}$. In the **second phase**, the adversary provides a message m and a nonce η. Based on the random bit b, x is returned, which is either an image containing m as watermark or an image generated from an initial latent $\mathcal{S}(r)$ for a distribution-preserving sampler $\mathcal{S}$ and a random seed r. Finally, $\mathcal{A}$ outputs a guess b' if x is watermarked or not; and the game checks whether this is correct ($b = b'$). Overall, a watermarking scheme is called IND$-CPA-secure with a security parameter n, if $\Pr[\mathsf{PrivK}^{\mathsf{IND\$-CPA}}_{\mathcal{A},\mathrm{WM}}(n) = 1] = \frac{1}{2} + \mathrm{negl}(n)$, where $\mathrm{negl}(n)$ is a negligible function, i.e., 2^{-n}.

An adversary $\mathcal{A}$ is called **nonce-respecting** if $\mathcal{A}$ never queries the same nonce multiple times. Note that a practical adversary usually has no control over the nonce if a protocol is designed securely. In case $\mathcal{A}$ is nonce-respecting, we can show that Gaussian Shading is secure and undetectable in

our new definition. We assume that ChaCha20 is IND$-CPA secure, as no non-generic attacks are known so far[1]. Formally, we get

$$\Pr[\mathsf{PrivK}^{\mathsf{IND\$-CPA}}_{\mathcal{A},\mathrm{GS}}(n) = 1] = \Pr[b = 1]\Pr[\mathcal{A}(x^{(m)}) = 1] + \Pr[b = 0]\Pr[\mathcal{A}(x) = 0] \quad (1)$$

$$= \frac{1}{2}\Pr[\mathcal{A}(\mathrm{GS}(k, \eta, m)) = 1] + \frac{1}{2}\Pr[\mathcal{A}(\mathcal{G}(\mathcal{S}(r))) = 0] \ . \quad (2)$$

On a real random input (second term), $\mathcal{A}$ cannot obtain any information and therefore just guesses with probability $\frac{1}{2}$. On a watermarked input (first term), $\mathcal{A}$ needs to recognize the output of ChaCha20 for an unknown key, which is hard by assumption. Therefore, the right hand side gets

$$= \frac{1}{2}(\frac{1}{2} + (q + 1)\,\mathrm{negl}(n)) + \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{2} + \mathrm{negl}(n) \ . \quad (3)$$

Note that Yang et al. show this behavior for an adversary $\mathcal{A}$ that has $q = 0$ queries. This adversary cannot use the same nonce multiple times as only one image is seen. However, in practice, users and thus attackers can generate multiple images. Our new definition holds for this case.

If we consider an adversary $\mathcal{A}'$ that is **not nonce-respecting**, there is an obvious attack. First, $\mathcal{A}'$ chooses $m^\star$ and $\eta^\star$ and passes this as (m_1, η_1) and as (m, η). $\mathcal{A}'$ obtains $x^{(m_1)}$ and x. Next, $\mathcal{A}'$ uses inversion[2] and the inverse sampler to recover the ciphertexts $c_1 = \mathcal{S}^{-1}(\mathcal{I}(x^{(m_1)}))$ and $c = \mathcal{S}^{-1}(\mathcal{I}(x))$. Idealized, if they both match[3], $\mathcal{A}'$ has found that this image is watermarked and outputs 1, otherwise 0. We compute the probability for this attacker and find that

$$\Pr[\mathsf{PrivK}^{\mathsf{IND\$-CPA}}_{\mathcal{A}',\mathrm{GS}}(n) = 1] = \Pr[b = 1]\Pr[\mathcal{A}'(x^{(m)}) = 1] + \Pr[b = 0]\Pr[\mathcal{A}'(x) = 0] \quad (4)$$

$$= \frac{1}{2}\Pr[\mathcal{A}'(\mathrm{GS}(k, \eta, m)) = 1] + \frac{1}{2}\Pr[\mathcal{A}'(\mathcal{G}(\mathcal{S}(r))) = 0] \quad (5)$$

$$= \frac{1}{2} \cdot (1) + \frac{1}{2}(1 - \mathrm{negl}(n)) \quad (6)$$

$$= 1 - \frac{1}{2}\,\mathrm{negl}(n) \ . \quad (7)$$

Clearly, $\mathcal{A}'$ has a non-negligible success probability—which is in fact close to 1 even with just one watermarked image—and can therefore easily distinguish between an unwatermarked image and a watermarked one. However, if the watermarked image is distinguishable from an unwatermarked one, given previous watermarked images, then the distribution of consecutively generated watermarked images differs from the original distribution obtained from unwatermarked z_T's.

In summary, if we do not choose a new nonce for every generated image, it is easy to distinguish these images from a random one as their latents are highly similar. However, if we alter the nonce for every image, the watermark stays hidden.

Note that PRC (Christ & Gunn, 2024) fulfills the same security definition. It incorporates the nonce in the generation of their encrypted message c in the generation process in a specified way and draws a new nonce for every image.

3.2 Key Management

Our novel proof of lossless performance extends the watermark usage to the realistic application scenario where multiple images are watermarked. The proof also shows how the encryption parameters k and η need to be specified—which has not been done for Gaussian Shading in its original publication. In the following, we compare the recommended cryptographic configuration with other configurations of how Gaussian Shading is currently deployed. In addition to security considerations, we

[1]Note that no real world symmetric cipher fulfills that definition in a strict sense. Nevertheless, the best known attacks require $\mathcal{O}(2^{n/2})$ time Boura & Naya-Plasencia (2023) which we consider infeasible for any practically relevant attack. Therefore, it can only guess a key with negligible success probability or needs to observe the encryption for one of its q requested nonces.

[2]Note that $\mathcal{A}'$ could even use a proxy model for inversion instead of the original model (Müller et al., 2024).

[3]Usually, they will not be exactly the same due to error in the recovery. However, they are close enough such that recovery is possible, i.e., $c \approx c_1$.

Configuration	FID ↓	CLIP ↑	LPIPS ↑
No Watermark	$66.7459_{0.6793}$	$0.3311_{0.0387}$	$0.6295_{0.0812}$
Same key, new nonce	$66.6537_{0.7014}$	$0.3311_{0.0400}$	$0.6274_{0.0744}$
Same key, same nonce	$68.9729_{0.5114}$	$0.3314_{0.0393}$	$0.5472_{0.0823}$

Table 1: Comparison of image quality and variety of different Gaussian Shading configurations. Compared to pure generation without watermark, our recommended *same key, new nonce* configuration preserves quality and variety. Note that the *new key, new nonce* setup would lead to equivalent results and is thus omitted. See Figure 3 in Appendix B for additional images examples.

also consider the practical deployment in terms of efficiency and generation quality / variety. In the latter case, we empirically assess the impact of each configuration in Table 1. Our empirical setup is described in Appendix B. In our analysis, we also shortly discuss the configuration of PRC.

As our proof shows, the stream cipher needs to be used in a **same key, new nonce** configuration[4]. This means the provider creates a fixed key k once. Given a fixed user id m and the fixed key k, the provider has to use a new nonce η to control the sampling process for every generated image. This configuration fulfills the security definition from Section 3.1 and is the way how to securely use a semantic watermark such as Gaussian Shading. Table 1 also shows that watermarking in this configuration preserves the quality and variety of the generated images compared to pure diffusion without any watermark. However, from a storage and runtime perspective, this watermarking setup is quite inefficient as it does not scale with the number of users and generated images / used nonces. In a normal message exchange setting, the sender could transmit the unencrypted nonce together with the encrypted message c. Unfortunately, we cannot just append η to c, as we need to transmit it in a robust fashion and the redundancy is applied before. Repeating the bits after the encryption would make the scheme non-random and allow for an easy distinction in IND$-CPA security. Hence, the provider has to store every used nonce for all the images it ever generated and, for watermark verification, has to XOR the encrypted message $\hat{c}$ with every possible keystream, which in turn requires a decryption with every stored nonce. PRC (Gunn et al., 2024) solves the nonce problem of Gaussian Shading by embedding its nonce into the watermarked image in a robust way.

An equivalent way in the semantic watermarking setup is the **new key, new nonce** configuration. This is in fact how Gaussian Shading is implemented in the GitHub repository (Yang et al., Last visit: Feb. 2025). This configuration has no cryptographic benefits compared to the previous configuration, and only increases storage requirements due to the need to save every key in addition.

Finally, there is the **same key, same nonce** configuration. This was the configuration that the follow-up work PRC (Gunn et al., 2024) assumed for its Gaussian Shading baseline. This configuration is efficiently deployable, but clearly not secure. It is not nonce respecting, so that $\mathcal{A}'$ can easily distinguish between marked and unmarked images. Moreover, this configuration of Gaussian Shading reduces image quality and variability (see Table 1). As the inputs to encryption and sampling are always identical for each user, the initial latent noise vectors remain similar as well.

4 Conclusion

In this work, we revisit the cryptographic primitives of distribution-preserving semantic watermarks. We provide a novel, more general proof to show that a semantic watermark has a lossless performance. The proof also covers the realistic case where multiple images are generated. We also properly specify the encryption parameters and discuss the implications regarding security, efficiency, and image quality & variety. In summary, a semantic watermark needs to be deployed such that a new nonce is used for every generated image.

Unfortunately, this configuration makes Gaussian Shading rather inefficient to deploy. In contrast, PRC fulfills our novel proof too and can be efficiently deployed. Finally, we note that it has implemented Gaussian Shading in the unfavorable configuration that affects both the security and generation quality. A direction comparison between PRC and Gaussian Shading is still open.

[4]Note that the **new key, same nonce** configuration is equivalent, as nonce and secret key are interchangeable as far as our analysis is concerned.

ACKNOWLEDGEMENTS

This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2092 CASA – 390781972 and by the Ministry of Culture and Science of Northrhine-Westphalia as part of the Lamarr Fellow Network.

REFERENCES

Daniel J Bernstein et al. Chacha, a variant of salsa20. In *Workshop record of SASC*, volume 8, pp. 3–5. Citeseer, 2008.

Christina Boura and Maria Naya-Plasencia. *Symmetric Cryptography, Volume 2: Cryptanalysis and Future Directions*. Wiley, Newark, 1 edition, 2023. ISBN 1394256302.

Miranda Christ and Sam Gunn. Pseudorandom error-correcting codes. In Leonid Reyzin and Douglas Stebila (eds.), *Advances in Cryptology – CRYPTO 2024*, pp. 325–347, Cham, 2024. Springer Nature Switzerland.

Hai Ci, Pei Yang, Yiren Song, and Mike Zheng Shou. RingID: Rethinking tree-ring watermarking for enhanced multi-key identification. In *Computer Vision – ECCV 2024*, 2024.

Sam Gunn, Xuandong Zhao, and Dawn Song. An undetectable watermark for generative image models. arXiv:2410.07369, 2024. To appear in ICLR 2025.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Proccessing Systems (NeurIPS)*, 2017.

Nicholas J Hopper, John Langford, and Luis Von Ahn. Provably secure steganography. In *Advances in Cryptology—CRYPTO 2002: 22nd Annual International Cryptology Conference Santa Barbara, California, USA, August 18–22, 2002 Proceedings 22*, pp. 77–92. Springer, 2002.

Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. Chapman & Hall/CRC cryptography and network security. CRC Press, Boca Raton, Fla. [u.a, 2. ed. edition, 2015. ISBN 9781466570269.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 2014.

Andreas Müller, Denis Lukovnikov, Jonas Thietke, Asja Fischer, and Erwin Quiring. Black-box forgery attacks on semantic watermarks for diffusion models. *arXiv preprint arXiv:2412.03283*, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proc. of Int. Conference on Machine Learning (ICML)*, 2021.

Phillip Rogaway. Nonce-based symmetric encryption. In Bimal Roy and Willi Meier (eds.), *Fast Software Encryption*, pp. 348–358, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-25937-4.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. In *Advances in Neural Information Proccessing Systems (NeurIPS)*, 2023.

Zijin Yang, Kai Zeng, Kejiang Chen, Han Fang, Weiming Zhang, and Nenghai Yu. Gaussian shading: Provable performance-lossless image watermarking for diffusion models. In *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Zijin Yang, Kai Zeng, Kejiang Chen, Han Fang, Weiming Zhang, and Nenghai Yu. Gaussian shading repository. https://github.com/bsmhmmlf/Gaussian-Shading/tree/09c678fadc7545acf7be12647ddf2a5e66f6a9dc, Last visit: Feb. 2025.

Kevin Alex Zhang, Lei Xu, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Robust invisible video watermarking with attention. arXiv:1909.01285, 2019.

A Watermarking Process in Gaussian Shading

Figure 2 shows the generation and verification process in Gaussian Shading.

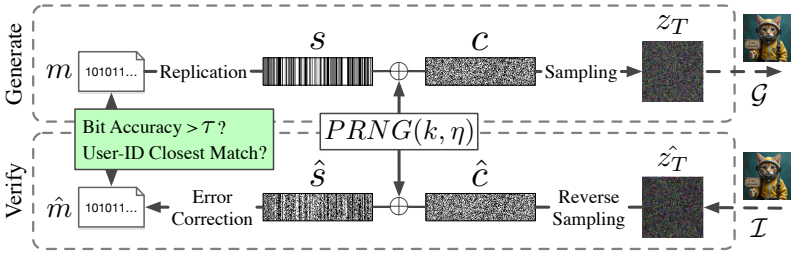

Figure 2: Overview over the watermark generation and verification process in Gaussian Shading

B Image Quality and Variety

Setup. To examine the impact of each Gaussian Shading configuration on image quality and variety, we use a setup similar to Gunn et al. (2024). We use Stable Diffusion 2.1[5] to generate 1.000 images from MS-COCO Lin et al. (2014) validation dataset captions and calculate FID (Heusel et al., 2017) with their ground truth images five times and report the mean value and standard deviation. We further report CLIP Scores (Radford et al., 2021) across all 5,000 generated images and their respective captions. In order to assess the variability of images, we generate 100 images from 10 diverse prompts from the prompthero website[6] and calculate pairwise LPIPS (Zhang et al., 2019) scores, similarly to the evaluation in (Gunn et al., 2024). We use the DPM sampler[7], as well as default guidance scale (7.5) and inference steps (50).

Visual Examples. Figure 3 shows the impact of Gaussian Shading configurations on image variety. While the same key, same nonce configuration leads to images with similar layouts, drawing new nonces for each image restores the image variety seen without watermarking.

[5]Stable Diffusion 2.1 Huggingface model card
[6]PromptHero webpage
[7]Huggingface DPMSolverMultistepScheduler documentation

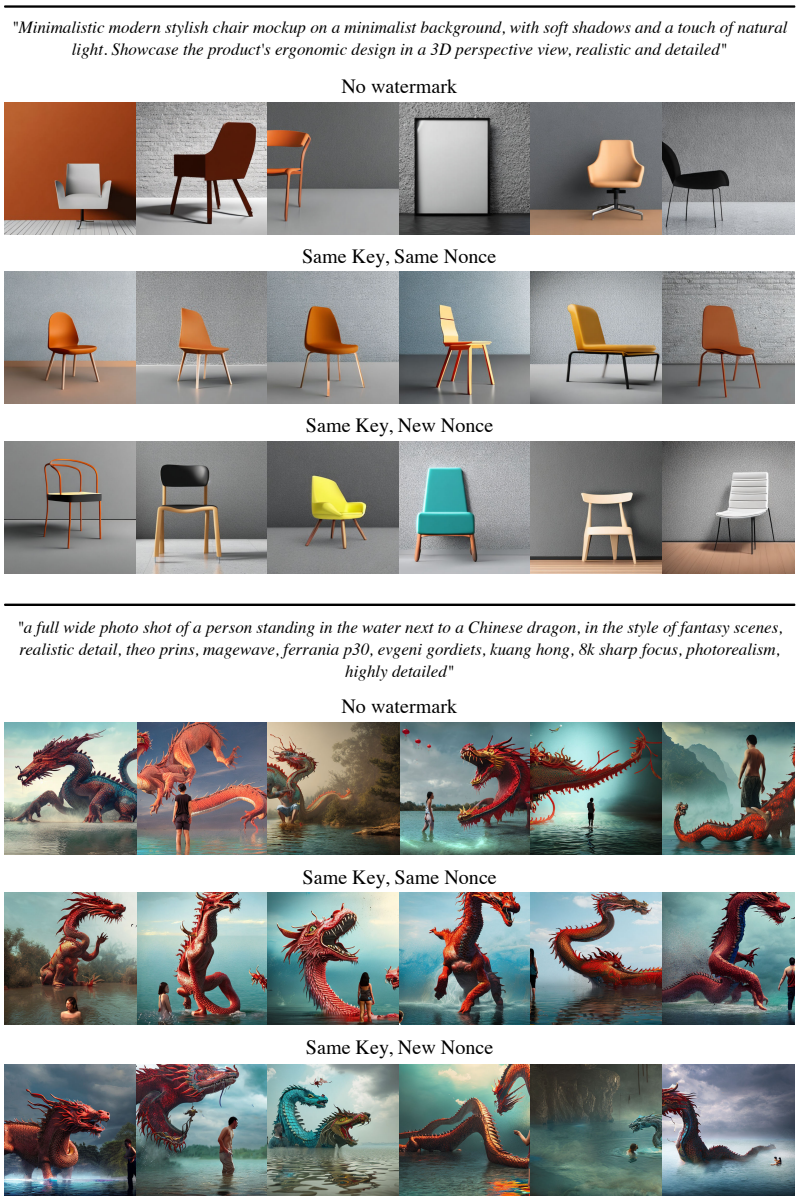

Figure 3: Impact of Gaussian Shading configurations on image variety.