Access Control for Diffusion Models by Random Masking the Covariance of Initial Noise Distribution

Temma Tanaka* and Kazuaki Nakamura*
* Tokyo University of Science, Japan
E-mail: 4625527@ed.tus.ac.jp. nakamura.kazuaki@rs.tus.ac.jp

***Abstract*—Recently, image generation models, particularly diffusion models, have advanced rapidly and are now widely used in various applications. On the other hand, these powerful models are also vulnerable to misuse for creating fake images. As a defense approach for fake image generation, methods for discriminating between real and generated data have been actively studied. However, since these discrimination models are generally trained on images generated by existing fake generators, they are fundamentally unable to maintain discrimination accuracy against new, unknown generation techniques. To address this issue, this paper proposes a novel, complementary defense framework that achieves access control for diffusion-based image generation models. Our framework requires a secret key to generate high-quality images and distributes it to only authorized users. Unauthorized users without the secret key can generate only low-quality images. To achieve this framework, we focus on the following characteristic of diffusion models: In the generation phase, we must use the same initial noise distribution as that in the training phase. Based on this characteristic, we make the initial noise distribution unpredictable by applying a random weight mask to its covariance, where the weight mask is distributed to only authorized users as the secret key. More specifically, we divide each image into fixed-size blocks and apply either of two weight values to the variance of all pixels in each block. In our experiment conducted on a face image dataset, the quality of generated images with the secret key achieves an FID score of less than 30. In contrast, the quality of generated images without the correct secret key is almost 300. These results demonstrate the effectiveness of the proposed method.**

I. Introduction

Artificial intelligence (AI) models for image generation have rapidly advanced in recent years and are now widely used in applications such as product mockups and web design. These AI models are downloadable and can be executed in local environments, suggesting further widespread adoption in the future. On the other hand, concerns have been increasing regarding the misuse of image generation models, such as the creation of fake images that contribute to the spread of disinformation or fraudulent activities. As the societal impact of these issues grows, the development of defense methods or countermeasures has become a pressing concern.

A direct approach to countering the generation of fake images is to develop a system capable of distinguishing between AI-generated and authentic images. Accordingly, a substantial body of research has been devoted to this challenge. For example, several methods have been proposed to classify real and generated face images using convolutional neural networks (CNNs) [1], [2], [3]. These approaches typically involve collecting large datasets of both real and generated images and training a classifier on these datasets. However, such classifiers generally rely on images generated by existing AI models. Consequently, these classifiers often struggle to maintain accuracy against new generative techniques that significantly differ in nature from those previously used. This fundamental limitation represents a key challenge of detection-based countermeasures.

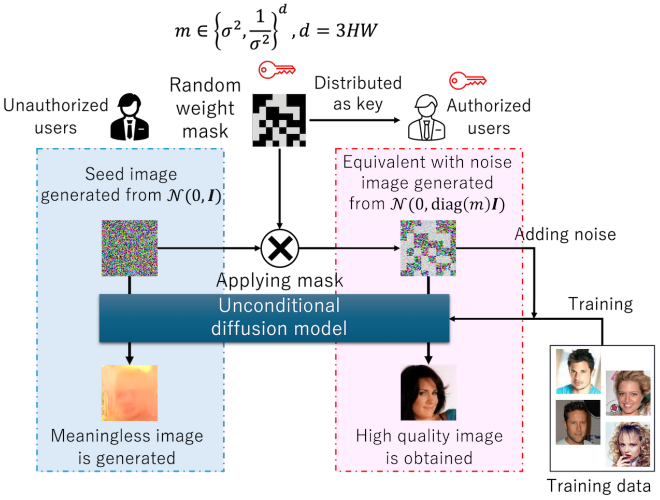

Fig. 1: Overview of proposed access control framework. We complicate the covariance of the initial noise distribution of a diffusion model by applying a random weight mask m to the diagonal of the covariance matrix. We distribute the mask m to only authorized users as a secret key.

To complement the above limitation, we consider integrating an access control mechanism into image generation models, particularly those working in local environments, to suppress their malicious use. In our proposed framework, service providers owning image generation models first distribute a secret key to authorized users. Then, the providers design an image generation model that can generate high-quality images only when the correct key is set; otherwise, the quality of generated images is intentionally degraded. This means that unauthorized (malicious) users who do not have the secret key cannot obtain high-quality images, as shown in Fig. 1. Although this framework does not perfectly eliminate the risk of misuse by authorized users, the authorization process itself serves as a deterrent against misuse. Furthermore, combining

979-8-3315-7206-8/25/$31.00

this framework with conventional detection-based countermeasures leads to a more robust defense. Note that similar access control mechanisms have been explored for image recognition models [4], [5], which inspired this work.

In this paper, we focus on diffusion models [6], [7], particularly unconditional diffusion models, as a specific class of image generation models. Diffusion models are capable of stably generating high-quality images and have recently gained significant attention. Hence, an access control framework for them is desirable. Our final goal is to extend the framework to conditional diffusion models that generate images from textual prompts. However, it is in the scope of our future work. The image generation process of diffusion models is as follows. First, an initial seed pattern is drawn from a known noise distribution. Then, the seed pattern is gradually denoised to obtain a high-quality image at the final step. In the above process, the standard normal distribution is commonly employed as the initial noise distribution, which is not a mandatory requirement. Therefore, in our proposed framework, we use a more complex normal distribution whose covariance is not easily guessable, where we use the covariance as a secret key. The contributions of this paper are:

- This is the first work to achieve an access control mechanism for diffusion-based image generation models.
- We propose a technique for utilizing the covariance of the initial noise distribution as a secret key.
- We experimentally demonstrate that the quality of images generated by our framework is drastically degraded without the secret key.

II. Related Work

A. Fake Image Detection

Fake image detection is closely related to liveness detection, a process to detect spoofed inputs in biometric authentication systems. For example, in camera-based face authentication systems, an attacker might conduct a spoofing attack by presenting a printed photo of an authorized user's face in front of a camera – known as a presentation attack. To counter this, liveness detection methods verify whether the presented data is from a live person or not. Some approaches leverage spatiotemporal features to distinguish between live inputs and spoofed ones [8]. In modern days, the emergence of AI-generated fake images has raised concerns over their use in spoofing attacks, leading to a growing body of research in fake image detection. For instance, Patel et al. proposed a method of detecting fake face images by identifying their distortions [9]. Conotter et al. exploited the fact that real human skin exhibits rapid, periodic color changes due to blood flow, and used the presence or absence of such fluctuations as a clue for fake detection [10]. Nguyen et al. focused on differences in texture complexity between real and fake faces and used these differences for detection [11]. More recently, numerous approaches based on convolutional neural networks (CNNs) have also been proposed. Raghavendra et al. employed fine-tuning of pretrained CNNs to detect fake images [1], while Nicolas et al. designed a customized pooling layer within the CNN architecture to improve detection performance [2]. Weize et al. proposed extending CNNs with new convolutional layers tailored for fake image detection [3].

Although these fake detection approaches have shown promising results, they typically rely on fake images generated using existing methods as training data. Hence, these detection methods may fail if a new image generation model produces fake images with characteristics significantly different from those seen in the training set. To tackle this limitation, this paper proposes a complementary approach based on access control, aiming to prevent unauthorized users from generating fake images. Rather than detecting fake images ex post facto, we focus on restricting access to image generators.

B. Access Control for Machine Learning Models

In the context of image recognition, access control techniques for AI models have been previously explored [4]. This framework encrypts all training images using a secret key and trains a classification model on the encrypted images. As a result, the trained model only produces correct recognition results when an input image is encrypted using the same secret key; otherwise, the model fails to recognize. The specific encryption technique employed in this framework is block-wise permutation and transformations (e.g., rotation and flipping), where the permutation order and the types of transformations are used as the secret key. Ito et al. extended this framework to apply it to semantic segmentation models [5]. Instead of encrypting the input images, their method encrypts intermediate feature maps. Specifically, they shuffle the channel order of a selected feature map based on a secret key. The segmentation model is trained on these shuffled features, enabling it to generate accurate segmentation maps in the test phase only when the correct key is used.

Although these access control methods provide a powerful tool for preventing the misuse of AI models, they cannot be directly applied to image generation models. This is because, unlike recognition models that take an image as input, image generation models produce images as outputs. To address this issue, this paper proposes a novel access control method for image generation models.

III. Proposed Method

A. Fundamentals of Unconditional Diffusion Models

We first summarize the fundamentals of unconditional diffusion models, which are necessary to understand our proposed method. A diffusion model consists of two processes: the forward process, which gradually adds noise to an image, and the reverse process, which progressively removes the noise from an initial seed pattern to generate a high-quality image. Fig. 2 illustrates these two processes. Both processes consist of T time steps. Let $\boldsymbol{x}_t$ be the image in the step t ($0 \leq t \leq T$). Note that this $\boldsymbol{x}_t$ is a d-dimensional vector obtained by flattening all the pixel values ($d = 3HW$ for an RGB image of size $H \times W$).

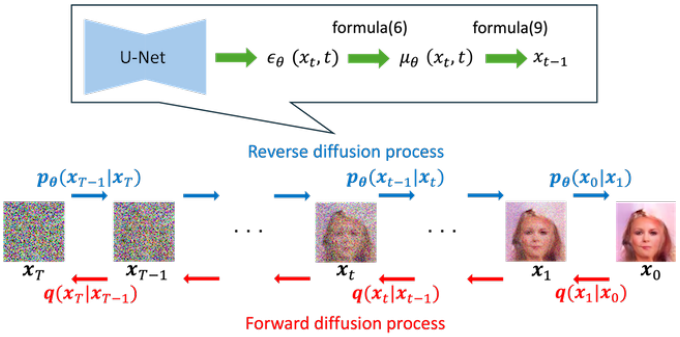

Fig. 2: Forward and reverse processes in diffusion models.

The forward process samples a noise pattern $\boldsymbol{\epsilon} \in \mathbb{R}^d$ from a noise distribution $p_{\text{train}}(\boldsymbol{\epsilon})$ and computes $\boldsymbol{x}_t$ as

$$\boldsymbol{x}_t = \sqrt{1-\beta_t}\boldsymbol{x}_{t-1} + \sqrt{\beta_t}\boldsymbol{\epsilon} \tag{1}$$

from $\boldsymbol{x}_{t-1}$, where β_t $(t = 1, \ldots, T)$ is a predefined constant known as the noise schedule. Various scheduling strategies have been proposed, including linear and cosine schedules [12]. Typically, a standard normal distribution is used as the noise distribution, namely, $p_{\text{train}}(\boldsymbol{\epsilon}) = \mathcal{N}(\mathbf{0}, \mathrm{I})$. In this setting, the distribution of $\boldsymbol{x}_t$ given $\boldsymbol{x}_{t-1}$ can be expressed as

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = \mathcal{N}\left(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t \mathrm{I}\right) . \tag{2}$$

By recursively multiplying this formula based on the Markov chain rule, we obtain

$$q(\boldsymbol{x}_t|\boldsymbol{x}_0) = \prod_{s=1}^{t} q(\boldsymbol{x}_s|\boldsymbol{x}_{s-1}) = \mathcal{N}\left(\boldsymbol{x}_t; \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0, (1-\bar{\alpha}_t)\mathrm{I}\right) , \tag{3}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$. This formula allows us to directly compute $\boldsymbol{x}_t$ as

$$\boldsymbol{x}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{x}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim p_{\text{train}}(\boldsymbol{\epsilon}) . \tag{4}$$

Unlike the forward process shown above, the reverse process $p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$ is not a normal distribution. However, if β_t is sufficiently small, we can approximate it as

$$p(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t) = \mathcal{N}\left(\boldsymbol{x}_{t-1}; \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t), \frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t \mathrm{I}\right) , \tag{5}$$

where the mean of the normal distribution, $\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t)$, is estimated from $\boldsymbol{x}_t$ and t by a U-Net-based neural network with a parameter θ. In practice, instead of directly estimating $\boldsymbol{\mu}_\theta$, the U-Net predicts the noise $\boldsymbol{\epsilon}$ as $\boldsymbol{\epsilon} = \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)$ and computes $\boldsymbol{\mu}_\theta$ as

$$\boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) = \frac{\sqrt{\bar{\alpha}_{t-1}}}{\sqrt{\bar{\alpha}_t}}\left(\boldsymbol{x}_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right) . \tag{6}$$

In the training phase, to train the U-Net, a diffusion model generates $\boldsymbol{x}_t$ from an original training image $\boldsymbol{x}_0$ by the forward process (4), and then minimizes the MSELoss $L(\theta)$ between the true and predicted noise, i.e.,

$$L(\theta) = \left\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{x}_t, t)\right\|^2 . \tag{7}$$

In the generation phase, we first sample an initial seed pattern $\boldsymbol{x}_T$ as

$$\boldsymbol{x}_T = \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim p_{\text{test}}(\boldsymbol{\epsilon}) . \tag{8}$$

Then we recursively denoise it as

$$\boldsymbol{x}_{t-1} = \boldsymbol{\mu}_\theta(\boldsymbol{x}_t, t) + \sqrt{\frac{1-\bar{\alpha}_{t-1}}{1-\bar{\alpha}_t}\beta_t}\boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim p_{\text{test}}(\boldsymbol{\epsilon}) \tag{9}$$

based on Formulas (5) and (6) to obtain a high-quality image $\boldsymbol{x}_0$. Note that, in the case of $t = 1$, we exceptionally calculate $\boldsymbol{x}_0$ as $\boldsymbol{x}_0 = \boldsymbol{\mu}_\theta(\boldsymbol{x}_1, 1)$.

To ensure high-quality image generation, the test-time noise distribution p_{test} should be exactly the same as the initial noise distribution p_{train} employed in the training phase. We leverage this property to achieve access control.

B. Overview of Proposed Access Control Method

As mentioned above, high-quality image generation using a diffusion model requires $p_{\text{test}}(\boldsymbol{\epsilon})$ to be identical to $p_{\text{train}}(\boldsymbol{\epsilon})$. In other words, if $p_{\text{test}}(\boldsymbol{\epsilon})$ differs from $p_{\text{train}}(\boldsymbol{\epsilon})$, the reverse diffusion process fails to properly denoise an initial seed pattern $\boldsymbol{x}_T$, making it difficult to generate high-quality images. Therefore, if we can construct a situation in which only authorized users know $p_{\text{train}}(\boldsymbol{\epsilon})$ while unauthorized users cannot obtain $p_{\text{train}}(\boldsymbol{\epsilon})$, we can achieve a desriable access control framework, where only authorized users can generate high-quality images. To this end, we propose to use not the standard normal distribution $\mathcal{N}(\mathbf{0}, \mathrm{I})$ but a more complex distribution that is difficult to guess, as the initial noise distribution $p_{\text{train}}(\boldsymbol{\epsilon})$.

However, the forward process shown in Formula (3) is formulated under the assumption that $p_{\text{train}}(\boldsymbol{\epsilon})$ follows a zero-mean normal distribution. Hence, distributions that do not meet this condition cannot be used. Taking this into account, the proposed method defines $p_{\text{train}}(\boldsymbol{\epsilon}) = \mathcal{N}(\mathbf{0}, \Sigma_{\text{train}})$, where its covariance matrix Σ_{train} is designed to hardly be guessable. The information required to reproduce Σ_{train} is treated as a secret key. By distributing this secret key only to authorized users, they can reproduce $p_{\text{test}}(\boldsymbol{\epsilon})$ that is identical to $p_{\text{train}}(\boldsymbol{\epsilon})$, allowing them to generate high-quality images. In contrast, unauthorized users who do not have the correct key cannot set $p_{\text{test}}(\boldsymbol{\epsilon})$ appropriately and are thus limited to generating low-quality images.

C. How to Set a Secret Key

As in standard diffusion models, we assume that Σ_{train} is a d-dimensional diagonal matrix. However, unlike the standard setting, our method sets either of two different values to all the diagonal elements of Σ_{train}. This setting means that we determine the noise intensity of $\boldsymbol{x}_T$ based on a pixel-wise and channel-wise varying distribution, because the diagonal elements correspond to the variance of the noise component for each pixel and channel. Specifically, we divide each training image into fixed-size blocks (of size $n \times n$ pixels), and assign either of two different values, $\frac{1}{\sigma^2}$ or σ^2, to each block. The assigned value is used as the variance of the noise component for all the pixels in the block. We perform this process separately for each channel, namely R, G, and B.

Sampling a noise pattern from such a distribution $\mathcal{N}(\mathbf{0}, \Sigma_{\text{train}})$ can be achieved as follows. First, we draw a noise pattern $\boldsymbol{\epsilon}$ from the standard normal distribution. Then, we define a random weight mask $m \in \left\{\frac{1}{\sigma^2}, \sigma^2\right\}^d$, which has the same size and the number of channels as the training images, and compute element-wise multiplication between $\boldsymbol{\epsilon}$ and $\sqrt{m}$. The mask m is generated randomly once at the beginning of the training process and remains fixed for all subsequent training and inference. This is equivalent to setting $\Sigma_{\text{train}} = \text{diag}(m)\text{I}$, where $\text{diag}(m)$ denotes the diagonal matrix whose diagonal elements are m. (See also Fig. 1 on the first page.) In the training phase, we train the U-Net using the noise distribution $p_{\text{train}}(\boldsymbol{\epsilon}) = \mathcal{N}(\mathbf{0}, \Sigma_{\text{train}}) = \mathcal{N}(\mathbf{0}, \text{diag}(m)\text{I})$. After training, we distribute the weight mask m only to authorized users as the secret key. With this key, authorized users can reproduce the same noise distribution $p_{\text{test}}(\boldsymbol{\epsilon}) = \mathcal{N}(\mathbf{0}, \text{diag}(m)\text{I})$ and generate high-quality images in the generation phase. On the other hand, unauthorized users cannot obtain $p_{\text{test}}(\boldsymbol{\epsilon})$ identical to $\mathcal{N}(\mathbf{0}, \text{diag}(m)\text{I})$ and thus are only able to generate low-quality images.

Due to the lack of the secret key, unauthorized users would use the standard normal distribution to sample an initial seed pattern $\boldsymbol{x}_T$. Pixel values in such a $\boldsymbol{x}_T$ are totally independent pixel-by-pixel. In contrast, initial seed patterns drawn from $\mathcal{N}(\mathbf{0}, \text{diag}(m)\text{I})$ tend to have a local similarity between values of neighboring pixels thanks to the block-wise assignment of noise variance. This difference facilitates a poor denoising performance without the secret key. Note that the length of the secret key is equal to the number of blocks, i.e., $N = 3 \times (H/n) \times (W/n)$. Since each block can take one of two values, the total number of key candidates is 2^N. Hence, dividing each image into smaller blocks increases the security level of the secret key. However, the effect of the block size on image quality is not immediately obvious; a smaller block size might lead to higher image quality even without the key. We experimentally evaluate this point in the next section.

The reason why we use $\frac{1}{\sigma^2}$ and σ^2 as the candidates of noise variance is to ensure numerical stability. In general, neural networks exhibit the most stable numerical behavior when their input values are around ± 1. If both candidate values are either smaller or greater than 1, the resulting pixel values in an initial seed pattern are less likely to be centered around ± 1. To avoid this, we employ the above two values, which always satisfy $\frac{1}{\sigma^2} < 1 < \sigma^2$ when $\sigma > 1$.

IV. Experiments

A. Experimental Setup

To evaluate the effectiveness of the proposed method, we conducted experiments on a face image dataset; specifically, we trained a face image generation model using the proposed method and verified how the quality of the model's generated images is affected by the presence or absence of a secret key. The specific scenarios we verified are:

- *Correct-key scenario*: the case of using the correct weight mask m in the generation phase.
- *Incorrect-key scenario*: the case of using a weight mask randomly chosen from 2^N candidates, which is not identical to the correct m. (Correct value of noise variance parameter σ is known.)
- *Without-key scenario*: the case of using the standard normal distribution as $p_{\text{test}}(\boldsymbol{\epsilon})$.

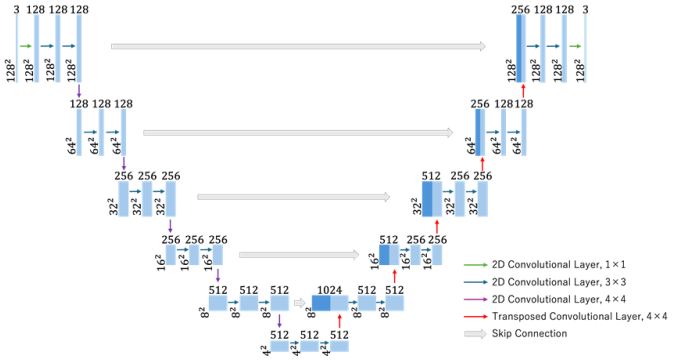

Fig. 3: Network architecture of the face image generation model (U-Net) used in our experiments. The first and last convolutional layers equipped with 1×1 kernels are used to adjust the number of channels. All other convolutional layers adopt 3×3 kernels, between which residual connections and Linear Attention are applied. The downsampling and upsampling layers are implemented using convolutional and transposed convolutional layers, respectively, both with a kernel size of 4×4 and a stride of 2.

The face dataset used in this experiment is CelebA, from which we randomly selected 16000 images as training data for a diffusion-based face image generation model. The network architecture of the trained model is shown in Fig. 3. For the training setting, we set the batch size to 16 and the number of training epochs to 2500. For hyperparameter setting, we set the block size n to $n = 2$ and the noise variance parameter σ to $\sigma = 3$.

To evaluate the quality of the generated images, we employed two metrics: Frechet Inception Distance (FID) [13] and face detection accuracy. The former, FID, measures the distance between two distributions of image features, one from real images and the other from test images. A smaller FID value indicates better quality of the test images. In our experiments, we used the training set of the U-Net as the real images for FID computation to test the same number of generated images. Lower FID values for the correct-key scenario and higher FID values for the incorrect-key scenario indicate better performance of the proposed method. Note that we failed to compute FID values in the without-key scenario due to the quite low diversity in the generated images. For the latter, face detection accuracy, we employed it as an evaluation metric because low-quality face images are expected to be missed by face detectors. Based on this assumption, we randomly selected 320 samples from the generated images in each scenario and tried to detect a face in each selected image. The face detector employed for this purpose is Google's Me-

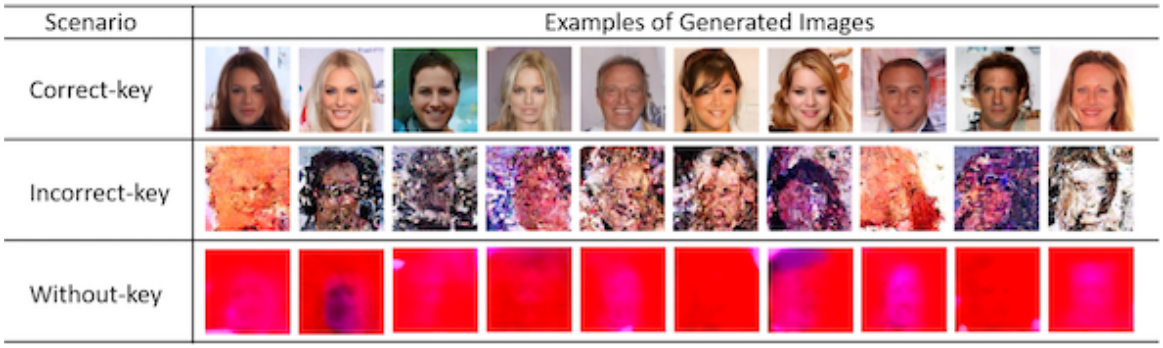

Fig. 4: Examples of generated images in each scenario. (num. of training images: 16000, $n = 2$, and $\sigma = 3$)

TABLE I: FID score and face detection accuracy of generated images in each scenario. (case of using 16000 training images)

Scenario	FID score	Face detection accuracy
Correct-key	27.390	98.4%
Incorrect-key	299.681	0.9%
Without-key	–	0.0%

diaPipe Face Detection [14]. A higher face detection accuracy for the correct-key scenario and a lower detection accuracy for the incorrect-key and without-key scenarios means better performance of the proposed method.

B. Results and Discussions

Fig. 4 depicts some examples of the generated images in each scenario. As shown in this figure, the proposed method can successfully generate high-quality images that are sufficiently clear to be recognized as human faces when the correct secret key is available. In contrast, in the incorrect-key scenario, the generated images lack clarity and cannot be recognized as a human face. Besides, in the without-key scenario, most of the generated images are nearly uniform in color and do not contain any facial structures at all. These results indicate that the proposed method produces a significant difference in image quality depending on the presence or absence of a secret key, strongly supporting its effectiveness.

The superiority of the proposed method is also quantitatively demonstrated in Table I, which shows the FID score and face detection accuracy in each scenario. In terms of the FID score, a sufficiently low value is obtained in the correct-key scenario, while a more than 10 times larger value is obtained in the incorrect-key scenario. Besides, in terms of face detection accuracy, the correct-key scenario achieves very high detection accuracy, while those in the incorrect-key and without-key scenarios are almost 0%. These results indicate that the quality of generated images is drastically degraded without the correct secret key.

C. Impact of Hyperparameters

To examine the impact of hyperparameters on the performance of the proposed method, we varied the block size n and the noise variance parameter σ as follows, and tested the

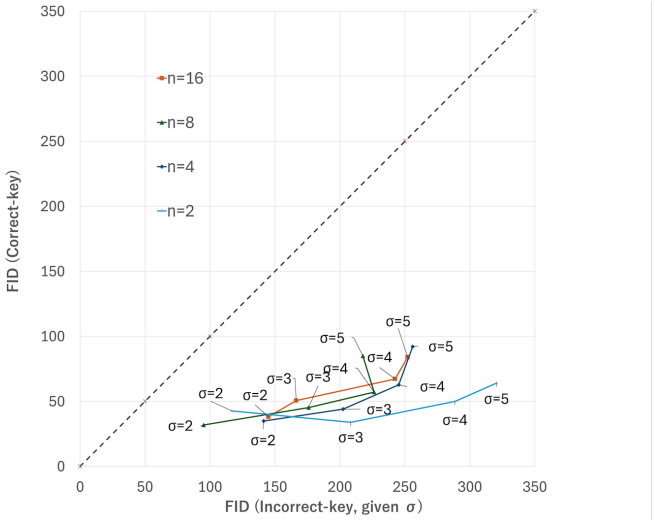

Fig. 5: FID scores in correct-key and incorrect-key scenarios under various n and σ. (case of using 3200 training images)

FID score and face detection accuracy in the correct-key and incorrect-key scenarios: $n = 2, 4, 8, 16$ and $\sigma = 2, 3, 4, 5$. Fig. 5 and Fig. 6 show the results of the FID scores and face detection accuracy, respectively. These results are not necessarily consistent with those shown in Table I, because we use only 3200 images for training a diffusion-based face image generation model to reduce the training time in this test.

In Fig. 5, the vertical axis shows the FID in the correct-key scenario, and the horizontal axis shows the FID in the incorrect-key scenario. Lower values on the vertical axis and larger values on the horizontal axis, i.e., nearer the bottom-right corner, indicate better performance. We can see from this figure that FID scores are consistently lower in the correct-key scenario across all settings, supporting the effectiveness of the proposed method. In general, smaller values of σ lead to better FID scores; however, when $\sigma = 2$, the FID score remains high even in the incorrect-key scenario, weakening the robustness of the proposed method. On the other hand, when $\sigma = 5$, the FID score is relatively high even in the correct-key scenario, meaning the quality degradation of the generated images. These results suggest that $3 \leq \sigma \leq 4$ offers a good

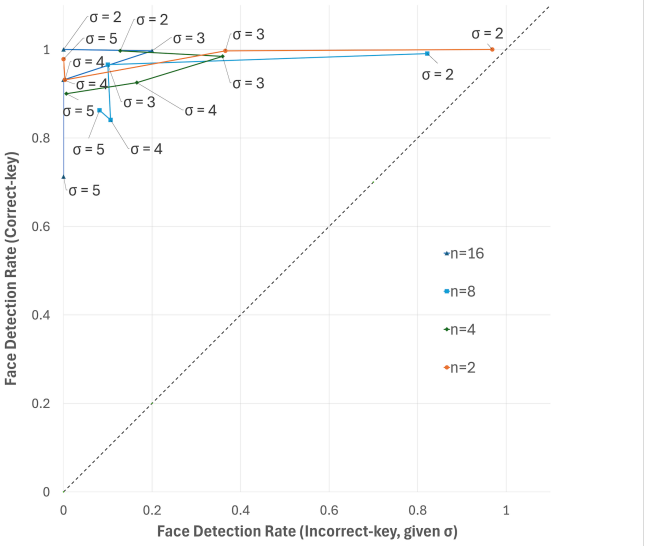

Fig. 6: Face detection accuracies in correct-key and incorrect-key scenarios under various n and σ. (case of using 3200 training images)

trade-off between quality and robustness. For the block size, changing n has little clear impact on the FID score in both scenarios. This fact indicates that using smaller blocks can increase the key length without compromising image quality and robustness of the proposed method.

In Fig. 6, the vertical and horizontal axes show the face detection accuracy in the correct-key and incorrect-key scenarios, respectively. The upper-left region in this figure, which corresponds to higher accuracy in the correct-key scenario and lower accuracy in the incorrect-key scenario, means better performance. In almost all cases, the face detection accuracy of higher than 90% is obtained in the correct-key scenario. However, when σ is too small, i.e., when $\sigma = 2$, high detection accuracy is sometimes obtained even in the incorrect-key scenarios. These results again support $\sigma = 3$ as the best balance. For the block size n, smaller blocks tend to result in relatively high face detection accuracy in the incorrect-key scenario. This is because images generated with an incorrect key tend to have particle-shaped noise whose size is the same as the block size n; larger particle noise prevents the face detection process more. We believe that this phenomenon is unique to face image generation models. Hence, the optimal block size should be decided according to another criterion, such as the FID score.

V. Conclusion

In this paper, we have proposed an access control method for diffusion-based unconditional image generation models. Our proposed method enables only authorized users with a secret key to generate high-quality images, while unauthorized users are limited to producing low-quality outputs. To this end, we leverage the principle that the generation phase of diffusion models requires using the initial noise distribution identical to that of the training phase. Specifically, we apply a random weight mask to the covariance of the initial noise distribution to complicate it, and distribute the weight mask only to authorized users as a secret key. By this technique, only authorized users can reproduce the initial noise distribution and obtain high-quality images. To evaluate the proposed method, we experimentally tested how the presence or absence of the secret key affects the quality of generated images. In the results, we obtained FID scores of less than 50 with the secret key and those of larger than 150 without the correct secret key by properly tuning the hyperparameters. This result demonstrates the effectiveness of the proposed method. Future work includes investigating the robustness of the proposed method against accidental use of similar weight masks. We also plan to examine the applicability of the proposed method to conditional diffusion models (those generating images from text prompts) and latent diffusion models like Stable Diffusion.

This work was supported by JSPS KAKENHI Grant Number JP25K03121.

References

[1] R. Raghavendra, K. B. Raja, S. Venkatesh, and C. Busch, "Transferable deep-cnn features for detecting digital and print-scanned morphed face images," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition Workshop*, 2017, pp. 1822–1830.

[2] N. Rahmouni, V. Nozick, J. Yamagishi, and I. Echizen, "Distinguishing computer graphics from natural images using convolution neural networks," in *Proc. IEEE Int'l Workshop on Information Forensics and Security*, 2017, pp. 1–6.

[3] W. Quan, K. Wang, D.-M. Yan, and X. Zhang, "Distinguishing between natural and computer-generated images using convolutional neural networks," *IEEE Trans. on Information Forensics and Security*, vol. 13, no. 11, pp. 2772–2787, 2018.

[4] M. AprilPyone and H. Kiya, "Privacy-preserving image classification using an isotropic network," *IEEE Trans. on Multimedia*, vol. 29, no. 2, pp. 23–33, 2022.

[5] H. Ito, M. AprilPyone, S. Shiota, and H. Kiya, "Access control of semantic segmentation models using encrypted feature maps," *APSIPA Trans. on Signal and Information Processing*, vol. 11, no. 1, pp. 1–8, 2022.

[6] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Int'l Conf. on Neural Information Processing Systems*, 2020, pp. 6840–6851.

[7] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *Proc. Int'l Conf. on Learning Representations*, 2021, pp. 1–22.

[8] U. Muhammad, Y. Zitong, and J. Komulainen, "Self-supervised 2d face presentation attack detection via temporal sequence sampling," *Pattern Recognition Letters*, vol. 156, pp. 15–22, 2022.

[9] K. Patel, H. Han, and A. K. Jain, "Secure face unlock: Spoof detection on smartphones," *IEEE Trans. on Information Forensics and Security*, vol. 11, no. 10, pp. 2268–2283, 2016.

[10] V. Conotter, E. Bodnari, G. Boato, and H. Farid, "Physiologically-based detection of computer generated faces in video," in *Proc. IEEE Int'l Conf. on Image Processing*, 2014, pp. 248–252.

[11] H. Nguyen, H. Nguyen-Son, T. Nguyen, and I. Echizen, "Discriminating between computer-generated facial images and natural ones using smoothness property and local entropy," in *Proc. Int'l Workshop on Digital Forensics and Watermarking*, 2015, pp. 39–50.

[12] T. Chen, "On the importance of noise scheduling for diffusion models," arXiv preprint arXiv:2301.10972, 2023, https://arxiv.org/abs/2301.10972.

[13] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *Proc. Int'l Conf. on Neural Information Processing Systems*, 2017, pp. 6629–6640.

[14] Google AI Edge, "Mediapipe face detector," https://ai.google.dev/edge/mediapipe/solutions/vision/face_detector/python, accessed: 2025-06-27.