future internet

MDPI

Article

Facial Privacy Protection with Dynamic Multi-User Access Control for Online Photo Platforms

Andri Santoso [1,*], Samsul Huda [2], Yuta Kodera [1] and Yasuyuki Nogami [1,*]

[1] Graduate School of Environmental, Life, Natural Science and Technology, Okayama University, Okayama 700-8530, Japan; yuta_kodera@okayama-u.ac.jp
[2] Green Innovation Center, Okayama University, Okayama 700-8530, Japan; shuda@okayama-u.ac.jp
* Correspondence: andri@s.okayama-u.ac.jp (A.S.); yasuyuki.nogami@okayama-u.ac.jp (Y.N.)

Abstract: In the digital age, sharing moments through photos has become a daily habit. However, every face captured in these photos is vulnerable to unauthorized identification and potential misuse through AI-powered synthetic content generation. Previously, we introduced SnapSafe, a secure system for enabling selective image privacy focusing on facial regions for single-party scenarios. Recognizing that group photos with multiple subjects are a more common scenario, we extend SnapSafe to support multi-user facial privacy protection with dynamic access control designed for online photo platforms. Our approach introduces key splitting for access control, an owner-centric permission system for granting and revoking access to facial regions, and a request-based mechanism allowing subjects to initiate access permissions. These features ensure that facial regions remain protected while maintaining the visibility of non-facial content for general viewing. To ensure reproducibility and isolation, we implemented our solution using Docker containers. Our experimental assessment covered diverse scenarios, categorized as "Single", "Small", "Medium", and "Large", based on the number of faces in the photos. The results demonstrate the system's effectiveness across all test scenarios, consistently performing face encryption operations in under 350 ms and achieving average face decryption times below 286 ms across various group sizes. The key-splitting operations maintained a 100% success rate across all group configurations, while revocation operations were executed efficiently with server processing times remaining under 16 ms. These results validate the system's capability in managing facial privacy while maintaining practical usability in online photo sharing contexts.

Keywords: facial privacy protection; selective facial encryption; multi-user access control; deep-learning applications; online photo platform

Academic Editor: Weizhi Meng and Christian D. Jensen

Received: 7 February 2025
Revised: 1 March 2025
Accepted: 7 March 2025
Published: 11 March 2025

Citation: Santoso, A.; Huda, S.; Kodera, Y.; Nogami, Y. Facial Privacy Protection with Dynamic Multi-User Access Control for Online Photo Platforms. *Future Internet* **2024**, *17*, 124. https://doi.org/10.3390/fi17030124

1. Introduction

Online platforms have transformed communication by facilitating the global sharing of personal content, such as photographs, through social networks and cloud storage. Although these platforms offer significant benefits for convenience and connectivity, they also pose substantial privacy risks [1]. Shared content frequently includes sensitive facial data, exposing individuals to potential privacy threats in the public domain. For instance, facial recognition technologies can exploit this data for biometric-based surveillance [2].

Biometric surveillance involves the collection and analysis of biometric data, such as facial images, to identify and monitor individuals [3]. Since facial images inherently carry the risk of unauthorized identification, their widespread availability on online platforms

increases privacy concerns, especially when combined with other data sources, such as online activities or location metadata [4].

In addition, analyzing facial images alongside contextual elements, such as background details, can inadvertently reveal sensitive information, including an individual's location or daily routines. These vulnerabilities have been exploited in real-world incidents, where perpetrators used facial details from social media to locate and harm individuals [5].

Furthermore, these concerns have become increasingly urgent due to recent advances in deep-learning generative models, particularly their ability to create and manipulate realistic facial data. For example, Karras et al. [6] introduced StyleGAN, a model that can generate highly realistic human faces. Similarly, Chen et al. [7] developed a face-swap technique that can seamlessly replace faces in both images and videos. Xu et al. [8] further proposed a framework for generating videos of people speaking from a single static image, demonstrating the rapid progression in video synthesis.

Despite their impressive capabilities, these technological advancements pose significant threats to individual privacy and societal integrity due to potential misuse, such as impersonation and misinformation. Although progress has been made in the detection of synthetic content [9–11], the rapid advancement of generative models continues to challenge existing detection methods as they generate increasingly realistic images that are often indistinguishable from authentic ones, even by human observers. This ongoing challenge underscores the urgent need for proactive privacy protection strategies that safeguard image data at its source, rather than relying solely on reactive measures to mitigate potential harms.

However, despite these growing concerns, many users continue to underestimate the privacy implications of sharing facial images online [12]. This lack of awareness is intensified by the ease of capturing and disseminating images on modern online platforms, resulting in a surge of biometric data available for exploitation [13].

While existing online platforms implement system-level privacy controls to protect biometric data, these measures remain inadequate in addressing facial data privacy challenges. Images shared online are often stored in unencrypted formats and accessible via direct links without authentication, making them vulnerable to unauthorized access and misuse. Furthermore, the persistence of files on online platforms poses another risk, as deleting photos does not guarantee the removal of facial data. Images may still be accessible via their URLs, even after deletion [14]. These limitations highlight the need for file-level facial privacy protection to ensure that facial data remain secure regardless of the image's online accessibility.

A common approach to file-level privacy in online platforms involves obscuring faces with emoticons or stickers. While this approach may be sufficient for concealing facial features at the file level, it lacks the sophistication of techniques such as image encryption, which enable the restoration of the original face when necessary.

Image encryption provides a robust solution to protect facial data by ensuring that the original face remains hidden until decrypted. Although encryption protects against unauthorized access, it introduces challenges in balancing an image's security and usability. For instance, encrypting an entire image can render it unusable for general viewing [15]. This tradeoff is particularly significant on online platforms, where the goal is to share visually meaningful images [16]. Full-image encryption undermines this usability by obscuring both facial and non-sensitive content, limiting the effectiveness of image sharing.

To address these limitations, our previous work introduced SnapSafe [17], a single-party system designed to protect individuals' privacy by selectively encrypting facial regions in images while preserving the visibility of non-sensitive content. SnapSafe focuses on proactive privacy protection at the source, ensuring that individuals' privacy is preserved

before an image enters the public domain. Using a deep-learning model based on the YOLOv8 architecture [18] for face detection and an AES algorithm for image encryption, SnapSafe demonstrated the feasibility of safeguarding facial data with minimal impact on image usability. This approach provides a practical solution for protecting facial data in shared images, particularly in the SnapSafe context, i.e., organizational settings where privacy concerns are paramount.

However, in the context of SnapSafe, the original system was designed primarily for organizational use cases, where a single entity—the organization—controlled both encryption and decryption. Consequently, only the organization could reveal faces, as it retained exclusive control over the decryption keys. This centralized model restricted access to the original faces solely to the organization.

Nevertheless, the demand for secure photo sharing extends beyond organizational contexts to online photo platforms, where images are shared among multiple users. Given the limitations of SnapSafe, particularly its single-party design and lack of dynamic access control mechanisms, our current work addresses these gaps by introducing dynamic access control mechanisms that enable controlled sharing of facial data among multiple users while ensuring flexible, user-centered privacy protection.

In this study, we classify users into two categories: image owners and image consumers. The interaction between these two groups, including how images are shared and accessed, is illustrated in Figure 1.

An image owner, referred to as Alice, is an individual who captures and uploads a photograph to the system, rather than the person depicted in it. This classification aligns with U.S. Copyright Office Circular 42 [19], which states that copyright belongs to the photographer, not the subject, unless transferred to another party. However, this assumption must be applied carefully, as publishing images containing identifiable individuals raises privacy concerns that vary across jurisdictions.

Once an image is uploaded, it may be shared publicly, making it accessible to the public domain, or within a restricted group, where access is limited to specific users. As shown in Figure 1, people who have permission to view facial data are referred to as image consumers, Bob representing an example. Bob can view the facial data unless his access is revoked, whereas the public domain can only access images with encrypted face regions.

Our proposed system enables image owners to encrypt facial data when necessary. This encryption may be optional for individuals prioritizing privacy or mandatory for organizations (e.g., schools, Non-Governmental Organizations (NGOs), or companies) to comply with regulations such as the General Data Protection Regulation (GDPR) [20]. For example, a school that shares images of its activities online acts as the image owner and can implement measures to protect students' privacy before publication. If a parent (acting as an image consumer) wishes to view their child's face in an image, they must submit a permission request, which the school administrator may approve or deny based on institutional policies and privacy considerations. However, the process of verifying image consumers—such as determining whether or not the requester is the child's parent in the school use case and whether or not they have the right to view the face—is beyond the scope of this study.

Upon approval, the system splits the original secret key into two shares: one assigned to the owner and the other to the consumer. For decryption, the image consumer must have access to both key shares, ensuring security and restricting access exclusively to authorized users. As part of this process, the image consumer must provide relevant details, such as the image identifier, to initiate the decryption. If all conditions are met, the system decrypts the facial data and displays the original face to the consumer. Although Figure 1 illustrates

the decryption process for the consumer, it does not depict the decryption process by the image owner, which will be detailed in subsequent sections.

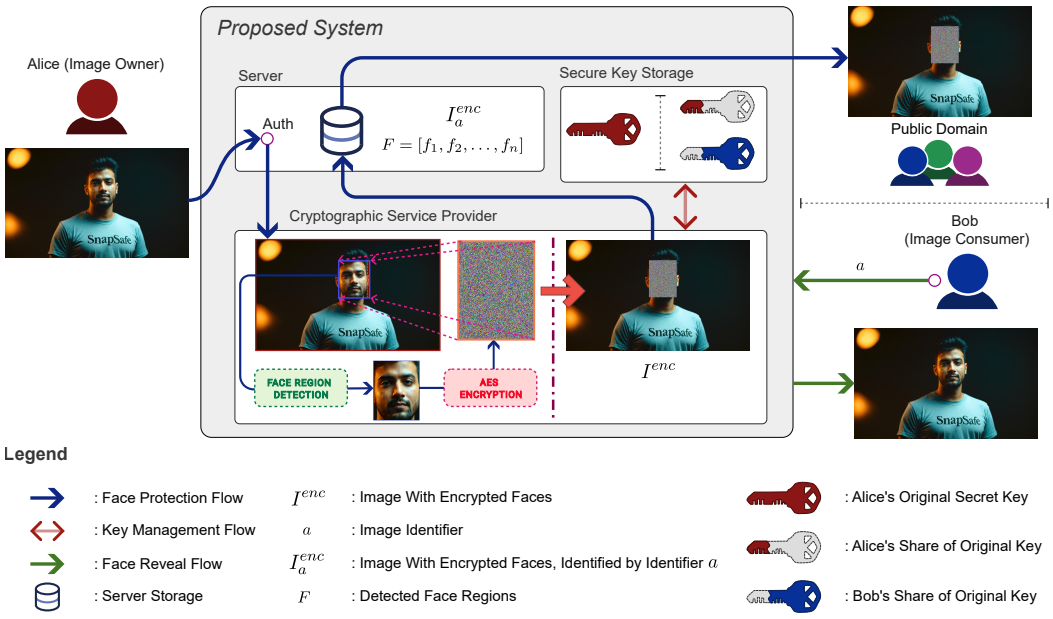


Figure 1. Overview of the interaction between image owners and consumers in the proposed system.

Additionally, the system enables the image owner to revoke previously granted access at any time, ensuring that full control over facial data is retained. This owner-centric permission system, combined with the request-based mechanism for initiating permissions, guarantees that non-facial areas of images remain accessible for general viewing, while facial data are protected under strict, dynamic access control policies.

By implementing encryption-based access control, our system ensures file-level security for images, allowing only authorized users to access facial data. This approach is particularly relevant for securely sharing privacy-sensitive images while enforcing strict access control measures. While our discussion primarily focuses on school environments, similar concerns extend to other domains. For instance, companies may need to regulate access to event photos, NGOs might publish images from community projects while safeguarding individuals' identities, and individuals may wish to share photos from social gatherings while preserving the privacy of those depicted.

Building on this foundation, our approach redefines the role of organizational users in the original single-party SnapSafe system. Instead of a single-user control model, these users now participate equally in sharing protected photos online. This shift to a multi-user access model expands the system's applicability, supporting a wider range of use cases.

The following sections outline the proposed system in detail and are structured as follows. Section 2 presents related works on facial privacy protection. Section 3 provides a preliminary discussion of key concepts and technologies relevant to the proposed system, along with an overview of this work's contributions. Section 4 describes the proposed system in detail, covering its components, the workflow for key operations, and security considerations. Section 5 discusses the experimental setup and analysis. Finally, Section 6 summarizes the study's findings and conclusions and outlines potential directions for future research.

2. Related Works

Facial privacy protection has attracted significant attention in recent years due to the proliferation of facial data on online platforms and the increasing risks of misuse. Various

techniques have been proposed to address these challenges, each targeting different aspects of facial data protection. Some approaches focus on image modification to protect facial privacy, while others emphasize user consent and access control mechanisms.

In this section, we review several existing works on facial privacy protection, highlighting their key features and limitations. Table 1 presents a comparative overview of approaches focused on image modification, while Table 2 provides a comparison of approaches that prioritize user consent and access control.

Table 1. Comparative overview of proposed approaches with some existing works in facial privacy protection.

Approach	Key Features	Face Reversibility	Multi User Access Control
DIFP [21]	Diffusion-based facial privacy protection network	Reversible but not pixel-perfect	No
Diff-Privacy [22]	Privacy-preserving facial image generation	Reversible but not pixel-perfect	No
PluGeN4Faces [23]	Person's attribute manipulation	Irreversible	No
DM [24]	Irreversible anonymization of facial features	Irreversible	No
SnapSafe [17]	Selective facial encryption for single-party scenarios	Pixel-perfect and reversible	No
Proposed System	Dynamic multi-user access control for facial privacy protection	Pixel-perfect and reversible	Yes

In the context of image modification to protect facial privacy, You et al. [21] proposed the Diffusion-Based Facial Privacy Protection Network (DIFP), which employs diffusion models to generate photorealistic and privacy-preserving images. DIFP ensures the reversibility of protected images, enabling the restoration of the original faces when necessary. Similarly, He et al. [22] introduced Diff-Privacy, which produces visually realistic and diverse anonymized images while maintaining the ability to recover original identities as needed. Although both approaches ensure reversibility, the restored images are not identical to the original images at the pixel level.

Suwala et al. [23] proposed PluGeN4Faces, a method for decoupling facial attributes from identity in StyleGAN's latent space, allowing for precise attribute manipulation while maintaining image realism. Although not explicitly designed for privacy protection, this approach could be adapted to anonymize facial images by altering identifiable attributes, such as age, hairstyle, or beard. Yang et al. [24] introduced the Digital Mask (DM), which uses 3D reconstruction and deep-learning algorithms to irreversibly anonymize facial features while preserving disease-relevant attributes. Unlike DIFP and Diff-Privacy, both PluGeN4Faces and DM lack the capability to reverse to the original image. Nevertheless, these methods offer promising alternatives for generating privacy-preserving facial images.

Another key aspect of facial privacy protection is access control, which involves obtaining user consent and granting users control over the usage of their facial data. Xu et al. [25] introduced a system where social media users depicted in a photo are notified and allowed to participate in decisions about photo sharing to prevent privacy breaches. Similarly, Tang et al. [26] proposed an automatic tagging framework that restricts photo access based on the identities of tagged individuals. Although these approaches focus on

user consent and access control, they do not address the challenges of file-level protection through the selective encryption of facial data in multi-user scenarios. Moreover, their approaches require users to provide personal photos for face model training, which may raise privacy concerns.

Table 2. Comparative overview of proposed approaches with some existing works in user consent and access control for facial privacy protection.

Approach	Key Features	Relies on User's Private Photos	File-Level Protection
Xu et al. [25]	Photo-sharing consent system for privacy protection	Yes	No
Tang et al. [26]	Automatic tagging framework for photo access control	Yes	No
Proposed System	Dynamic multi-user access control for facial privacy protection	No	Yes

While the discussed approaches provide valuable insights into facial privacy protection, they do not address the challenges of selective image encryption in multi-user scenarios. Additionally, although methods such as DIFP and Diff-Privacy ensure reversibility, the restored images are not pixel-perfect, which limits their applicability in scenarios requiring precise image restoration. In contrast, our proposed system provides a comprehensive solution for dynamic multi-user access control in online photo platforms. It safeguards facial data through file-level selective image encryption while allowing pixel-perfect restoration when needed. Moreover, it eliminates the need for participating users to submit personal photos for face model training, enhancing both privacy and usability.

3. Preliminaries

3.1. YOLOv8 for Face Detection

Object detection models are essential for identifying objects in images and videos. One of the most popular object detection models is the YOLO (You Only Look Once) model, which is known for its remarkable speed and accuracy in object detection tasks. Over the years, YOLO has undergone multiple iterations, each aimed at improving the model's performance and efficiency.

YOLOv8 [18] is one of the recent iterations of the YOLO model, which is optimized for both speed and accuracy. Building upon YOLOv5 [27], it incorporates enhancements to refine the original architecture.

Figure 2 illustrates an overview of the YOLOv8 architecture, which comprises three primary components: the backbone, the neck, and the head. The backbone network extracts features from the input image, the neck network refines these features, and the head network performs a specific task, such as object detection, classification, segmentation, or pose estimation. The model is trained on extensive datasets to learn object features and accurately perform the designated task.

Although YOLO is designed primarily for general object detection tasks, its adaptability facilitates specialized applications. For example, training YOLOv8 on a dataset containing facial images enables the model to accurately detect facial regions. This capability is essential for facial data encryption, as it allows systems to identify and encrypt facial regions while leaving other parts of the image unmodified.

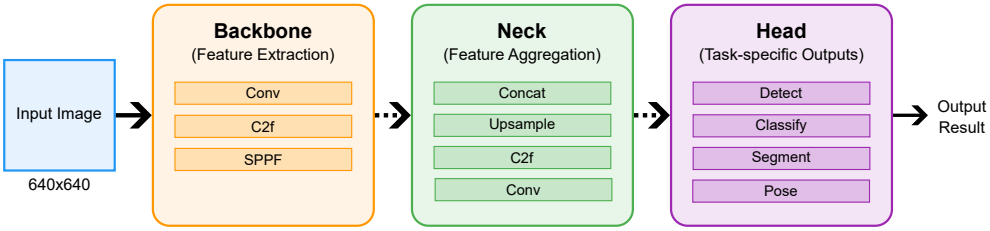


Figure 2. Overview of YOLOv8 architecture.

In this work, we leverage the YOLOv8 architecture for face detection to identify facial regions in images. By training the model on facial datasets, we achieve accurate face detection and selectively encrypt these regions to protect sensitive facial data. This approach ensures that facial data remain secure while maintaining the visibility of non-sensitive content in shared images, thereby preserving the usability of the image. The details of the model's training process and configuration are discussed in Section 5.1.2.

3.2. Selective Image Encryption

Selective image encryption differs from traditional methods, such as AES, by taking the image file's structure into account. Conventional encryption algorithms treat an image as a generic data stream, making the encrypted output unreadable by standard image viewers. Because these methods disregard the original file structure, necessary modifications can be easily performed before encryption to ensure the plaintext complies with algorithmic requirements. However, in selective image encryption, preserving the file structure is essential, making such modifications more challenging or even impractical.

For instance, the AES algorithm employs padding to adjust the plaintext size to match its required block size of 16 bytes. While this ensures compliance with AES encryption standards, it can disrupt the structure of image data, making the encrypted output incompatible with standard image encoding formats. This limitation underscores the need for specialized image encryption techniques that preserve structural integrity while maintaining usability in common image viewers.

Some existing image encryption methods produce encrypted images that remain viewable. Recent research has proposed various encryption techniques incorporating methods such as Hilbert curve scrambling, Discrete Wavelet Transform (DWT), and dynamic DNA coding to ensure pixel-level security [28]; fractal geometry combined with chaotic maps and Paillier homomorphic encryption [29]; and chaotic synchronization systems enhanced by Radial Basis Function Neural Networks (RBFNN) and Particle Swarm Optimization (PSO) [30]. However, these approaches typically encrypt the entire image.

While full-image encryption is essential for protecting sensitive data, there are scenarios where selective encryption of specific regions is more appropriate [17]. For example, in online photo sharing, users may prefer to encrypt only sensitive areas while leaving non-sensitive regions unaltered. Encrypting an entire image presents a challenge in balancing security and usability. Although the image remains viewable in a standard viewer, it appears as noise and is incomprehensible to the human eye. In contrast, selective encryption of an image encrypts specific regions while preserving the interpretability of the non-sensitive areas.

Our work in SnapSafe [17] introduced a method to selectively encrypt sensitive facial regions in an image, ensuring that facial data remain secure while the rest of the image remains usable for sharing. This approach is particularly suitable for online platforms where sharing visual information is essential while maintaining privacy protection.

However, implementing selective image encryption effectively requires addressing several challenges. First, encrypted regions must be compatible with reintegration

into the original image without disrupting its structure. Additionally, the approach should support encryption of rectangular regions of varying sizes, as sensitive areas in images can differ in dimensions and location. For example, the method proposed by Geng et al. [28] is unsuitable for this purpose due to its reliance on Hilbert curve scrambling and DWT decomposition, which only support square regions. This limitation makes it infeasible for encrypting arbitrary rectangular areas, such as those detected using object detection algorithms.

Our previous work [17] introduced an algorithm to overcome this challenge by adjusting the bounding box of the image segment prior to AES encryption. This adjustment ensures that the byte size of the region is a multiple of 16 bytes, eliminating the need for padding and facilitating seamless reintegration into the original image. Since the encrypted data maintain structural compatibility with standard image formats, this approach enables selective encryption without introducing compatibility issues.

In this work, we enhance the applicability of the selective image encryption approach introduced in SnapSafe by extending its functionality to support multi-user access control. Building on SnapSafe's foundation, our improved system integrates dynamic access control mechanisms, allowing multiple authorized individuals to securely access facial data while maintaining strong privacy safeguards. This advancement ensures that SnapSafe is no longer restricted to a single-entity access model, making it more versatile and practical for broader use cases.

3.3. Contributions

In this paper, we present a multi-user security system that facilitates the controlled sharing of facial data on online photo platforms. The proposed system extends SnapSafe to support multi-user scenarios, enabling the selective encryption of facial regions in images while preserving the visibility of non-sensitive content. By incorporating dynamic access controls, the system ensures that facial data remain secure and accessible only to authorized users. The key contributions of this work are as follows:

- Key Splitting for Access Control: The system introduces key splitting as a mechanism to enable access control over shared facial data. This approach allows the image owner to selectively share access to facial regions in shared images. This key splitting operation enables the secure sharing of facial data while keeping the image owner exclusive access to the original key.
- Owner-Centric Permission System: The system implements an owner-centric permission system, designating the image owner as the primary authority for granting access to facial data. This system ensures that the owner retains full control over the shared data, enabling the management of access permissions for multiple users. In addition to granting access, the owner can revoke previously granted permissions at any time, ensuring that facial data remain protected under the owner's supervision.
- Request-Based Access Mechanism: The system incorporates a request-based access mechanism that allows users to request permission to access protected facial regions. By default, this mechanism ensures that facial regions in images shared on online platforms remain protected. Only the image owner has access to these regions, while other users must submit access requests. The owner retains full authority to approve or reject such requests, thereby maintaining control over the visibility of facial data.

4. Proposed System

4.1. System Components

The system consists of three key components: the client application, the server, and the Cryptographic Service Provider (CSP). These components collaboratively enable the secure

sharing of facial data among multiple users. Each component serves a critical function, ensuring that facial data are protected and accessible only to authorized users. The following sections detail the roles and responsibilities of each component.

4.1.1. Client Application

The client application serves as the primary interface for users to interact with the system. It is responsible for collecting user input, such as credentials and original images, and securely transmitting these data to the server.

The client application does not perform cryptographic operations or store sensitive cryptographic information locally. It operates as a front-end interface that allows users to initiate cryptographic processes performed on the server. These processes include encrypting facial regions, key splitting for access sharing, and decrypting the facial regions only for authorized users.

Additionally, the client application provides an interface for user registration, login, and access management. Through this interface, users can register for accounts, log in to the system, and manage access permissions for shared images (including granting, rejecting, or revoking access). Furthermore, to maintain secure user sessions and ensure the integrity of communication with the server, the application employs JSON Web Tokens (JWT) for authentication.

4.1.2. Server

The server plays a central role in the system by handling user requests, verifying user credentials, and securely storing encrypted images along with metadata related to encryption processes. Furthermore, it acts as an intermediary, forwarding client requests related to cryptographic operations to the CSP, where those operations are performed. A detailed discussion of the CSP component is provided in Section 4.1.3.

In this context, communication between the server and the CSP is authenticated and conducted over a secure internal network, ensuring that only the server is exposed to the public internet. By serving as an intermediary, the server ensures the secure execution of cryptographic operations and protects sensitive data from unauthorized access.

Finally, the server manages user sessions through the use of JWT authentication, ensuring that users remain authenticated throughout their interactions with the system. By validating the JWT token in each request, the server can authenticate users without the need for server-side session storage.

4.1.3. Cryptographic Service Provider

The Cryptographic Service Provider (CSP) is responsible for executing all cryptographic operations and managing the associated cryptographic secrets. Specifically, it manages the generation of secret keys, the production of salts required for cryptographic processes, the encryption of facial regions in images, the implementation of key splitting for shared access, the decryption of facial regions for authorized users, and the secure storage of sensitive data.

The storage system within the CSP is designed to mitigate the risk of unauthorized access. This component utilizes authentication mechanisms to allow the server to perform various actions within the CSP, such as generating, storing, and retrieving keys, as well as salt management. Additionally, the CSP ensures that the server can access stored data only through authenticated and secure internal communication channels, thereby preventing unauthorized access. To further enhance security, the CSP encrypts sensitive secrets using the user's password, ensuring that only the correct password holder can decrypt the key. The implementation details of secure data handling in the CSP's storage system are discussed in Section 4.3.1.

4.2. System Workflow

The system involves two categories of users: the image owner and the image consumer. The proposed system facilitates interactions between these two user categories to enable the secure sharing of facial data in images. The image owner is responsible for encrypting images, granting access to selected consumers, and revoking previously granted access. The image consumer can request access to protected facial regions and view them upon the image owner's approval.

Furthermore, the system supports multiple image consumers (e.g., Bob, Carol, and Dave) requesting access to the same image. The image owner (e.g., Alice) retains full control over access permissions and can manage each relationship independently. For instance, Alice may grant access to Bob and Carol while denying it to Dave. Additionally, Alice can revoke access previously granted to Carol while maintaining Bob's access. This granular control ensures that the system restricts unauthorized users to non-facial areas of the image while permitting only authorized users to view protected facial regions.

These interactions are made possible through the collaboration of the system's primary components, as previously described: the client application, the server, and the CSP. Together, these components ensure the secure sharing of facial data by managing user interactions, processing requests, and performing cryptographic operations.

The following sections provide a detailed description of these interactions, as outlined in the system workflow for key operations: user registration, face encryption, face decryption by owner, key splitting, and face decryption by consumer. Each operation is illustrated in a step-by-step process, detailing the interactions between the system components and the roles of the image owner and image consumer.

4.2.1. User Registration

The user registration process begins when an individual (e.g., Alice, Bob, or Carol) creates an account by submitting credentials through the web client application. For example, as illustrated in Figure 3, Alice submits her username (*uname*) and password (*pwd*) to the server. Upon receiving these credentials, the server validates them and issues a JSON Web Token (JWT), which is then securely stored on Alice's web client. The JWT serves as a session token for subsequent requests, enabling secure communication between the client and the server.

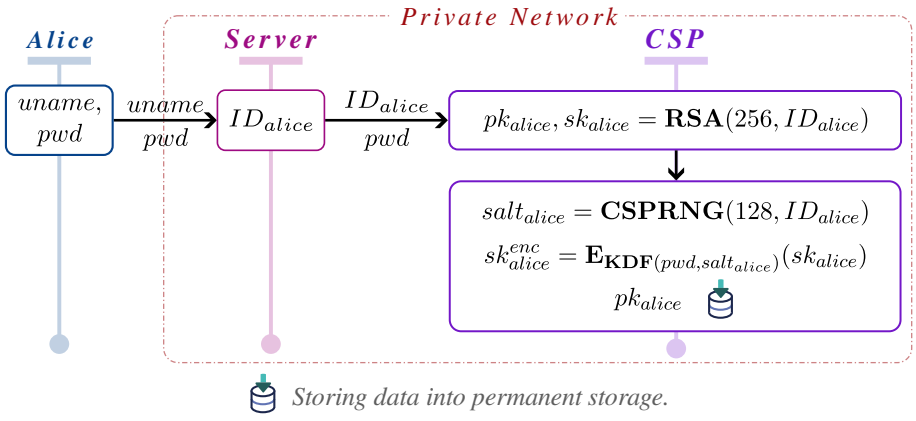


Figure 3. User registration process.

Subsequently, the server forwards Alice's registration request to the CSP, which generates a unique public–private key pair, denoted as pk_{alice} (public key) and sk_{alice} (private key). The key pair is generated using the $\mathbf{RSA}(256, ID_{alice})$ function, where 256 specifies the key size and ID_{alice} is a unique identifier associated with Alice. The ID_{alice}

serves as the key identifier that will be used to retrieve the stored key from the CSP's key storage management.

After generating the key pair, the CSP retains the public key (pk_{alice}) in plaintext while encrypting the private key (sk_{alice}) as sk^{enc}_{alice}. The encryption process employs the function **E** and a secret key derived from Alice's password (pwd) using a key derivation function (**KDF**). The **KDF** uses pwd and a randomly generated salt ($salt_{alice}$) to derive the encryption key. The salt is generated using a cryptographically secure pseudorandom number generator, $\mathbf{CSPRNG}(128, ID_{alice})$, where 128 denotes the salt's bit length. Finally, the CSP securely stores $salt_{alice}$, sk^{enc}_{alice}, and pk_{alice} in its secure key storage system.

This password-based encryption mechanism ensures that only Alice can access her private key. Even if the keys stored within the CSP are compromised, an attacker cannot retrieve sensitive facial data protected by the system.

4.2.2. Face Encryption

After completing the registration process, users can protect facial information in shared images through encryption. For instance, as illustrated in Figure 4, when Alice intends to share an image with a broad audience while ensuring the preservation of facial privacy, she uploads an image I containing facial regions via the client application to the server.

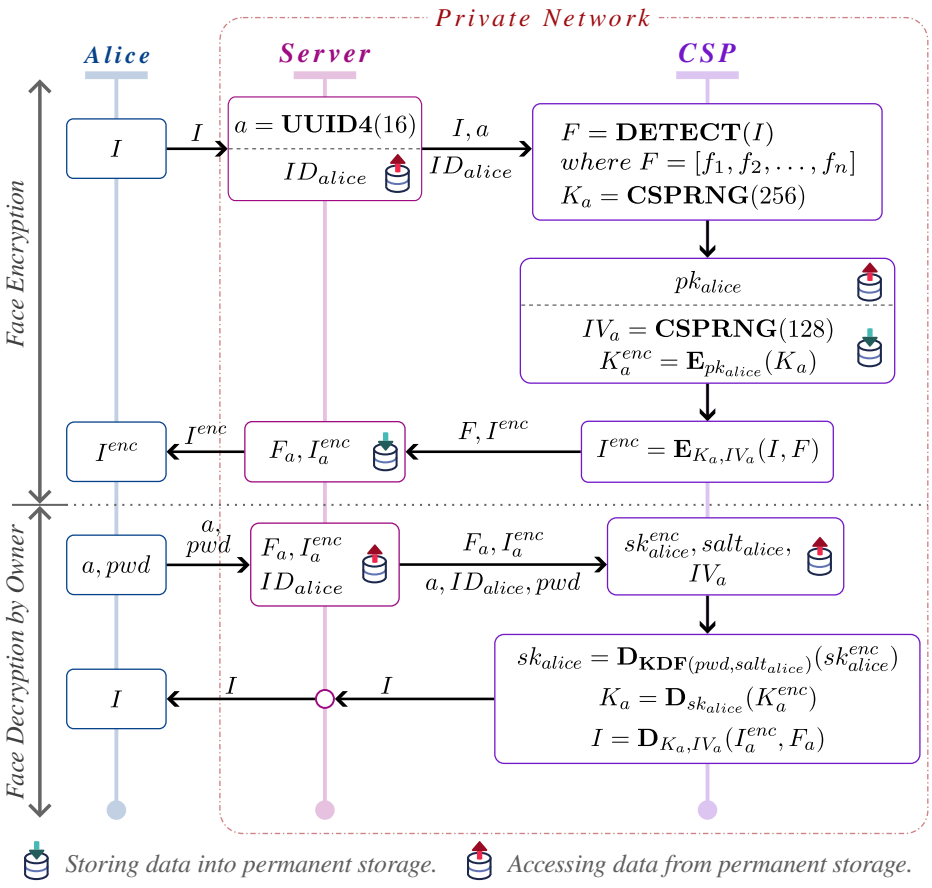


Figure 4. Face encryption and decryption by owner.

Upon receiving the request, the server validates it and generates a unique identifier, a, for image I using a UUID version 4 generator, denoted as $\mathbf{UUID4}(16)$, where 16 specifies the byte length. Subsequently, the server retrieves Alice's identifier, ID_{alice}, from storage and forwards the image I along with a and ID_{alice} to the CSP.

The CSP processes the image I to identify facial regions using the function $\mathbf{DETECT}(I)$, resulting in a set of facial regions, $F = \{f_1, f_2, \ldots, f_n\}$. Each $f_i \in F$ represents the bound-

ing box coordinates of the i-th detected face, defined as $f_i = (x_i, y_i, w_i, h_i)$, where (x_i, y_i) denotes the top-left corner coordinates and (w_i, h_i) specifies the width and height of the bounding box, respectively.

Building on our prior work [17], this face detection process is designed to ensure that the byte size of each facial region matches the 16-byte block size of AES, thus eliminating the need for padding to ensure the compatibility of encrypted data with the original image format. This approach enables seamless reintegration of encrypted facial data into the original image, preserving the image's structural integrity.

Subsequently, the CSP generates a symmetric key, K_a, and an initialization vector, IV_a, to encrypt the facial regions, F, within the image, I. Using these generated values (K_a and IV_a), the CSP encrypts the regions corresponding to F, resulting in the encrypted image I^{enc}. This encryption process follows the methodology outlined in our previous work [17], where facial regions are encrypted sequentially based on the order of f_i in F. This sequential approach ensures that even overlapping facial regions can be fully reconstructed during decryption.

Then, for key storage, the symmetric key K_a is encrypted using Alice's public key pk_{alice}, producing K_a^{enc}. The CSP stores K_a^{enc} and IV_a securely for future decryption.

Finally, the CSP returns the encrypted image I^{enc} and facial region metadata F to the server. The server associates these elements with identifier a to produce I_a^{enc} and F_a, storing them in persistent storage for future retrieval. Subsequently, the server transmits I^{enc} to the client application for display to Alice, completing the encryption process.

4.2.3. Face Decryption by Owner

To initiate the face decryption process, Alice, the image owner, must provide both the image identifier (a) and her password (pwd) to the server, as illustrated in Figure 4. Although Alice is authenticated with the server via a JSON Web Token (JWT) during registration or login, her password is required for each face decryption operation. This password is used to derive a decryption key, which is subsequently employed to decrypt Alice's private key stored on the server. This approach ensures that only Alice, as the sole holder of the password, can access her private key and decrypt the facial data.

Upon receiving and validating the request, the server retrieves Alice identifier ID_{alice}, the encrypted image I_a^{enc}, and associated face regions F_a from its storage and forwards them as a request to the CSP alongside Alice's password pwd.

Upon receiving the request from the server, the CSP then retrieves Alice's salt $salt_{alice}$, encrypted Alice's private key sk_{alice}^{enc}, and initialization vector IV_a from its storage. Using pwd and $salt_{alice}$, the CSP calls function $\mathbf{KDF}(pwd, salt_{alice})$ to derives a decryption key K_{dec}, which is then used to decrypt sk_{alice}^{enc} using function $\mathbf{D}$. This decrypted private key, sk_{alice}, is then used to decrypt K_a^{enc}, producing K_a. With K_a, IV_a, and F_a, the CSP decrypts face regions within I_a^{enc}, producing the original image I. The CSP then returns the processed image to Alice through the server. This process ensures that only Alice, as the image owner, can access and view the protected facial data.

4.2.4. Key Splitting for Shared Access

After encrypting the facial regions in an image I, Alice can selectively share the image with others, such as Bob, by granting access to the encrypted facial data. To initiate this shared access mechanism, Bob submits a request to Alice for permission to view the protected regions. Alice can then approve or reject the request. Upon approval, Bob is granted access to the encrypted facial regions in the image. This process is illustrated in Figure 5.

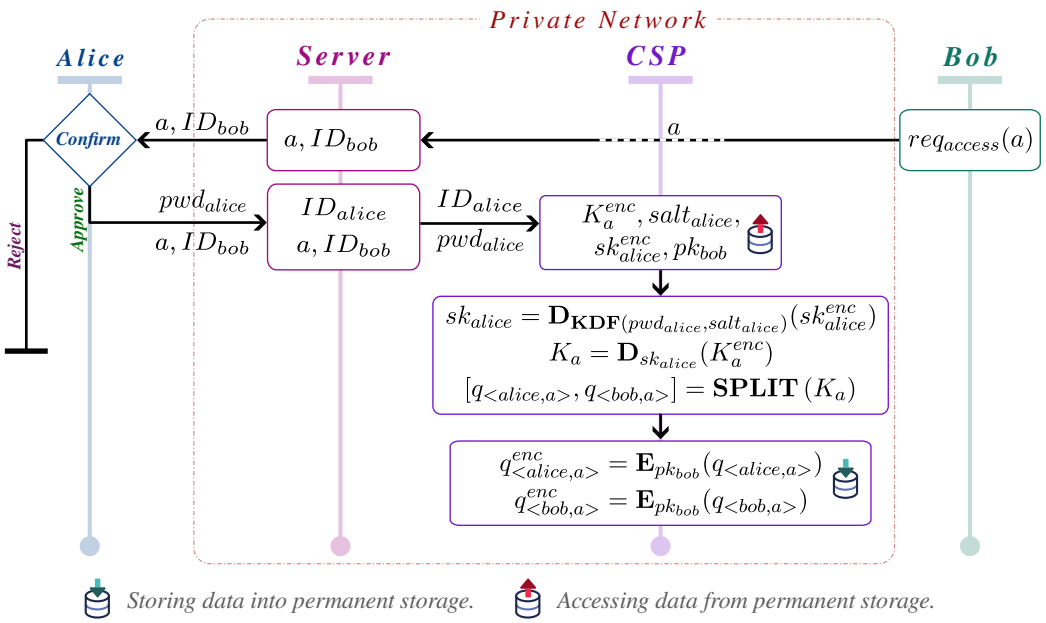


Figure 5. Key splitting process for shared access. This figure illustrates the process by which Alice splits the symmetric key of a shared image into two shares: one for herself and one for Bob, the image consumer.

The access-granting process begins when Bob sends an access request to Alice via the server. This request includes the image identifier a, which uniquely identifies the image. Upon receiving and validating the request, the server identifies that the user with identifier ID_{bob} is requesting access to the image and notifies Alice of the pending request.

If Alice rejects the request, she submits a request to the server to deny access, and no further action is taken by the server except for deleting the request. If Alice approves the request, she submits a follow-up request to the server to initiate the key-splitting process. This request contains the image identifier a, Alice's password pwd_{alice}, and Bob's user identifier ID_{bob}. The server validates Alice's request and forwards it to the CSP for processing.

Upon receiving the server's request, the CSP retrieves the encrypted symmetric key K_a^{enc}, the salt generated for Alice $salt_{\text{alice}}$, Alice's encrypted private key $sk_{\text{alice}}^{\text{enc}}$, and Bob's public key pk_{bob} from its storage. Since $sk_{\text{alice}}^{\text{enc}}$ is encrypted, the CSP decrypts it using a key derived from Alice's password pwd_{alice} and $salt_{\text{alice}}$, yielding sk_{alice}. This private key is then used to decrypt K_a^{enc}, resulting in the original symmetric key K_a, which is required to reveal the facial data in the image.

Once the symmetric key K_a is obtained, the CSP splits it into two shares: one for Alice $q_{\langle \text{alice},a \rangle}$ and one for Bob $q_{\langle \text{bob},a \rangle}$—using Shamir's secret sharing scheme, denoted as $\mathbf{SPLIT}(K_a)$. These shares are then encrypted using Bob's public key pk_{bob}, ensuring that only Bob can decrypt and recombine them. As Bob's public key is available in plaintext, this process does not require his direct involvement to provide credentials during key splitting. Finally, the encrypted shares are stored and can be accessed by Bob when he submits a subsequent request to the server to perform decryption of the facial regions, as described in the next section.

If Alice later decides to revoke Bob's access to the facial data, she can submit a revocation request to the server. Upon verifying Alice's identity and her ownership rights to the image, the server forwards the request to the CSP. The CSP then deletes Alice's share $q_{\langle \text{alice},a \rangle}$ of the key for image a from its key storage. Since both shares, $q_{\langle \text{alice},a \rangle}$ and $q_{\langle \text{bob},a \rangle}$, are required for the consumer to decrypt the facial data in image a, removing Alice's share $q_{\langle \text{alice},a \rangle}$ automatically renders Bob's share $q_{\langle \text{bob},a \rangle}$ invalid. This revocation

mechanism ensures that Alice retains full control over access to the facial data in her shared images, even after initially granting permission.

4.2.5. Face Decryption by Consumer

The decryption process for Bob, the image consumer, primarily follows similar steps to those performed by Alice, the image owner. The difference is in the method of accessing the symmetric decryption key. While Alice has direct ownership of the image and can access the symmetric key directly, Bob does not possess this direct ownership. Instead, Bob uses key shares provided by Alice during the key splitting process to reconstruct the original symmetric key. With the reconstructed key, Bob decrypts the encrypted facial regions in the image to access the protected data. This decryption process for the consumer is illustrated in Figure 6.

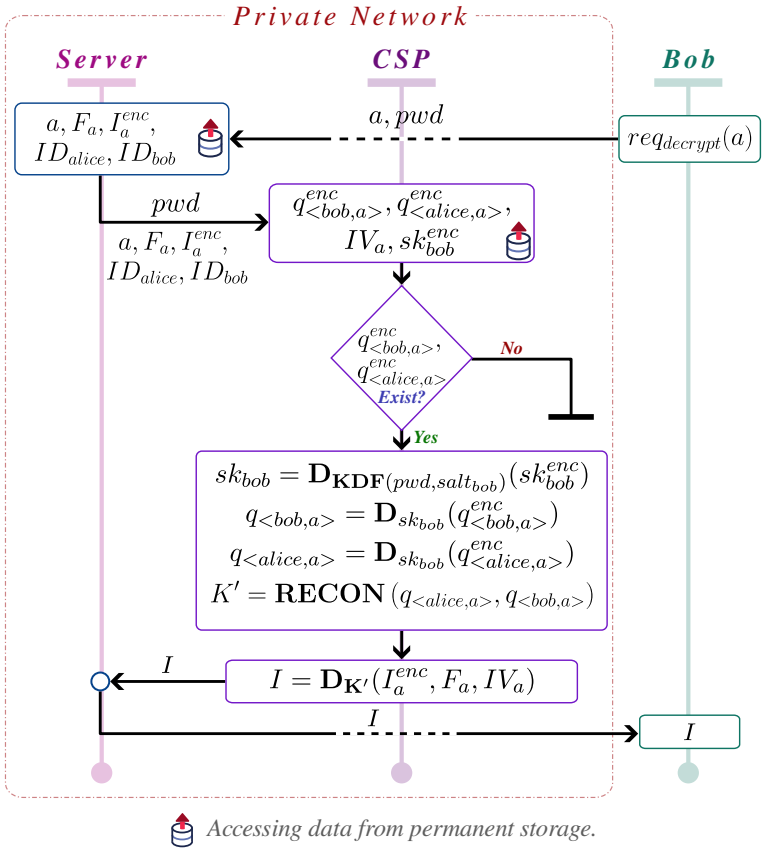


Figure 6. Face decryption by consumer: This figure illustrates the process by which an image consumer, such as Bob, decrypts facial regions in a shared image using key shares provided by the image owner, Alice.

To initiate the face decryption process, Bob sends a request to the server containing the image identifier a and his password pwd. Similar to the image owner, the password is essential for each decryption request to successfully reveal the facial regions.

Upon receiving and validating Bob's request, the server retrieves the face coordinates F_a, the encrypted image I_a^{enc}, along with the identifiers for Alice (ID_{alice}) and Bob (ID_{bob}) from its storage. This information is then forwarded to the CSP for processing.

Upon receiving the request from the server, the CSP retrieves the encrypted key shares ($q^{\text{enc}}_{\langle alice,a\rangle}$, $q^{\text{enc}}_{\langle bob,a\rangle}$), Bob's encrypted private key sk^{enc}_{bob}, along with the initialization vector IV_a used to encrypt the facial regions in the image I_a^{enc}. The CSP checks for the availability of both Alice's and Bob's key shares. If either share is missing, the decryption process terminates. If both shares are available, the CSP proceeds with key reconstruction.

To reconstruct the symmetric key for image decryption, the CSP decrypts Bob's private key sk_{bob}^{enc} using Bob's password pwd and a salt $salt_{bob}$ associated with Bob's account, yielding sk_{bob}. Once sk_{bob} is obtained, the CSP decrypts the encrypted key shares $q_{\langle alice,a\rangle}^{\text{enc}}$ and $q_{\langle bob,a\rangle}^{\text{enc}}$ using sk_{bob}. The decrypted key shares are then combined using the function $\textbf{RECON}(q_{\langle alice,a\rangle}, q_{\langle bob,a\rangle})$, which uses Shamir's secret sharing scheme to reconstruct the symmetric key K'. Using K' and IV_a, the CSP decrypts the facial regions in the image I_a^{enc}, producing the original image I. The CSP then returns the processed image to Bob via the server, completing the decryption process.

4.3. Security Considerations

4.3.1. Security Implementation for Data Handling

The system ensures the security of data in transit and stored data through multiple mechanisms. Data in transit is protected using the HTTPS protocol, which encrypts information exchanged between the client application and the server. This ensures that sensitive data, such as user credentials and image data, remain confidential during transmission. Additionally, the server communicates with the CSP through an internal network, isolating data exchanges from external threats and maintaining security.

To protect stored data, the system does not retain original images or sensitive cryptographic information on the server. Instead, the server stores only face-encrypted images and metadata related to the face encryption processes. Sensitive data, such as private and symmetric keys, are securely managed by the CSP using two primary mechanisms. First, the CSP leverages Hashicorp Vault [31], a dedicated key management system, to store secrets securely. Access to Vault is strictly controlled, and the CSP communicates with it using the AppRole authentication method [32], which enforces specific policies and login constraints to obtain tokens with the required permissions. Second, the CSP employs password-based encryption, where private keys are encrypted with a key derived from the user's password. This ensures that only the user can access their private key, even if the secrets management system is compromised. Finally, although public keys are not sensitive, they are also stored in Vault for consistency.

4.3.2. Potential Security Risks from Adversarial Image Attacks

While our proposed system implements a data protection mechanism for both data in transit and stored data, it is essential to consider potential adversarial attacks that could compromise image security. One such attack involves the use of manipulated images to bypass the face protection mechanism. Although these risks are concerning, they are unlikely to occur under normal circumstances. In our system, users do not typically attempt to bypass encryption, as they are the ones who choose to encrypt their data. If they wish to bypass encryption, they can simply opt not to encrypt their data.

However, the risks associated with manipulated images may arise in adversarial attack scenarios, such as when a user's device is compromised. In such cases, an attacker could alter images before they are uploaded to the server without the user's knowledge. This manipulation could cause the face detection algorithm to fail, rendering the face encryption mechanism ineffective.

To mitigate this risk, the system could implement additional security measures, such as image integrity verification. By verifying the integrity of uploaded images using cryptographic hashing algorithms, the system could detect unauthorized modifications. However, this approach may not be foolproof, as attackers could potentially manipulate both the image and its hash value to evade detection. Therefore, a comprehensive security strategy is needed to address this risk effectively. Nonetheless, since adversarial attacks are not

the primary focus of this work, a detailed analysis of their risks and potential mitigation strategies is beyond the scope of this study and intended for future research.

5. Experiments

This section presents our experimental validation of proposed system capabilities and performance. We evaluated the system's performance and effectiveness of face encryption, key splitting, face decryption operations, and revocation mechanism. The detail of the experiment design, including the system configuration, test dataset, and evaluation scenario, is provided in Section 5.1. The results of the experiments are presented and analyzed in Section 5.2.

5.1. Experiment Design

5.1.1. System Configuration

The experiments were conducted on a Ubuntu 24.04.1 LTS machine with the following specifications: 12th Gen Intel® Core™ i7-12700H, 16 GB RAM, 512 GB SSD, and NVIDIA GeForce RTX™ 3050 Ti Laptop GPU. The system was implemented using Docker containers running on Docker v27 [33] to ensure reproducibility and isolation. The following containers were deployed for the experiments:

- **Server Container:** Django v5.1.3, base image: python:3.10-slim, port: 8000
- **Database Container:** PostgreSQL v17, base image: postgres:17, port: 5432
- **CSP Container:** Django v5.1.3, base image: python:3.10-slim, port: 8081
- **Vault Container:** HashiCorp Vault v1.18.3, base image: hashicorp/vault:1.18, port: 8200

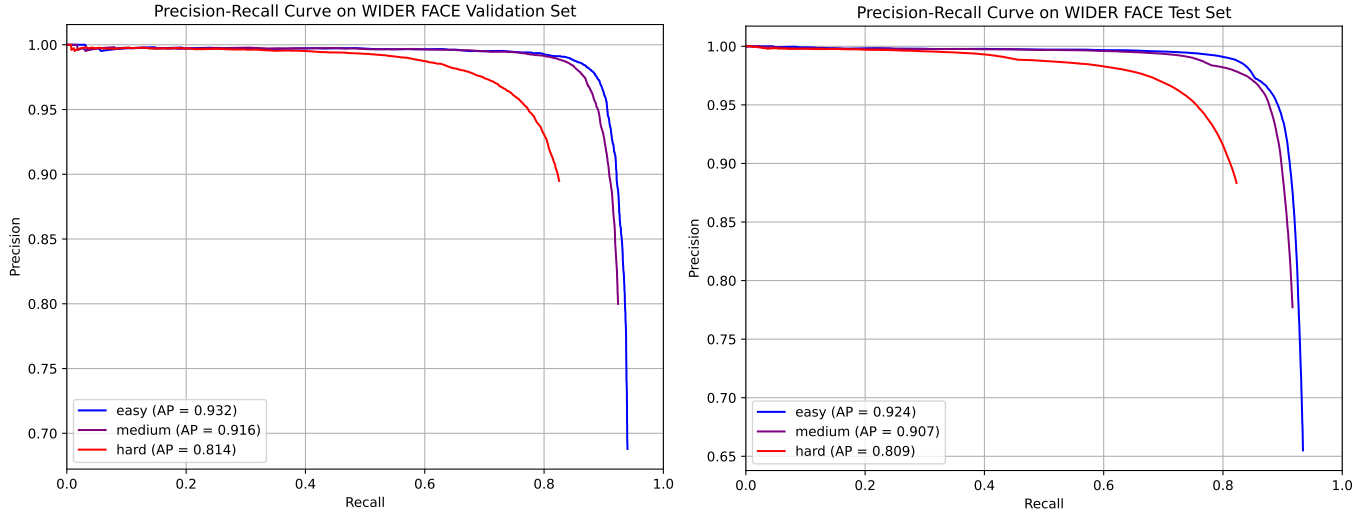
Only the server container is accessible from the public network. In this configuration, the server container communicates with both the CSP and the database containers through the internal network. Additionally, for key management operations, the CSP container interacts with the Vault container over the same internal network to ensure the secure management of cryptographic keys.

5.1.2. Deep-Learning Model Configuration

For face detection, we employed the same model as in our previous work [17], based on the YOLOv8 architecture. The model was trained on the WIDER Face dataset [34] to detect facial regions in images. Specifically, we utilized the YOLOv8m pre-trained model [35] and fine-tuned it on the WIDER Face dataset to adapt it to our use case. YOLOv8m is a medium-sized model in the YOLOv8 family, achieving a 50.2 mAP on the COCO validation set [36]. With 25.9 million parameters and 78.9 billion FLOPs, the model achieves a balance between accuracy and computational efficiency. YOLOv8m is more complex than YOLOv8n and YOLOv8s but less resource-intensive than YOLOv8l and YOLOv8x. It processes images in 234.7 ms on a CPU (ONNX) and 1.83 ms on an NVIDIA A100 GPU (TensorRT), making it suitable for real-time applications [37].

Our fine-tuned model was trained using Python version 3.10 with PyTorch version 2.2.0 [38] and the Ultralytics Python library version 8.1.9 [18]. Training was performed on a system with an NVIDIA RTX 3070 GPU for 300 epochs, achieving a precision of 0.875 and a recall of 0.651 in the final epoch.

As shown in Figure 7, our model achieved high accuracy on both the validation and test sets of the WIDER Face dataset. In the test set, it obtained Average Precision (AP) scores of 0.924 for easy, 0.907 for medium, and 0.809 for hard levels, demonstrating its effectiveness in detecting facial regions in previously unseen images, even in challenging conditions.



(**a**) (**b**)

Figure 7. Precision–recall curve of the fine-tuned YOLOv8m model on the (**a**) validation and (**b**) test sets of the WIDER Face dataset.

5.1.3. Test Dataset

In this study, we assessed the performance of the system using a subset of images from the WIDER Face dataset [34]. The WIDER Face dataset includes images with annotated facial regions, making it an ideal benchmark for evaluating face detection capabilities. It is widely recognized for its diverse collection of images, with varying levels of difficulty, categorized as easy, medium, and hard.

In this study, we selected a subset of images from the validation set of the WIDER Face dataset to assess the performance of the proposed system. This selection was made due to the unavailability of ground truth annotations for the test set. To ensure an accurate assessment of the system's computational performance, we focused on images where our deep-learning model could reliably detect all facial regions.

We selected the face regions from the 'medium' level of the WIDER Face validation dataset. Based on the detection results of our model, the selected images were categorized into four groups according to the number of faces detected: Single (1 face), Small (2–5 faces), Medium (6–10 faces), and Large (11–65 faces). An overview of these face categories and the corresponding number of images in each category is provided in Table 3.

Table 3. Face categories and image counts.

Face Category	Number of Faces	Number of Images
Single	1	996
Small	2–5	715
Medium	6–10	171
Large	11–65	118

5.1.4. Materials and Method

In our study, we used images from the WIDER Face validation set, as explained in Section 5.1.3, to evaluate the proposed system when the experimental evaluation considered only quantitative metrics. However, when the evaluation involved qualitative assessment with images depicting human faces, we used AI-generated images to simulate real-world environments while mitigating ethical and privacy concerns. These images were generated

for each face category listed in Table 3 and were later used in Section 5.2.3 to illustrate the results of the face encryption and decryption operations performed by our system.

We used the Stable Diffusion 3.5 Medium model from Stability AI [39], which comprises 2.5 billion parameters and generates images ranging from 0.25 to 2 megapixels. In our experiments, all generated images had a resolution of 1280 × 768 pixels, 96 dpi, and a bit depth of 24. For each face category, we repeatedly generated images from scratch until we obtained high-quality, natural-looking depictions of real-world environments.

The images were generated using specific prompts that defined the number of people, their attributes, and the surrounding conditions. Some prompts precisely generated the requested number of individuals, while others did not. However, the generated images contained a sufficient number of individuals to satisfy the requirements of each face category. Details of the prompts and hyperparameters used to generate the test images for each category can be found in Appendix A.

We generated four images in total, each corresponding to a different face category. The AI-generated images were initially saved in PNG format and later converted to JPG before being processed by our system for evaluation.

5.1.5. Evaluation Scenario

The system facilitates interaction between two primary roles: the image owner and the consumer. Image owners can upload images, view associated facial data, and manage access privileges by granting or revoking permissions for consumers. Conversely, consumers can request access to view facial data within images shared by image owners, with their access dependent on the owner's approval.

During evaluation tests, users were assigned to one of the following roles:

- **Owner:** The image owner can view facial data in their shared images.
- **Consumer:**
  - **Without Access:** Unauthorized users cannot view facial data in shared images.
  - **After Grant:** Authorized users can view facial data in shared images after the owner grants access.
  - **After Revoke:** Previously authorized users lose access to facial data in shared images after the owner revokes access.

In the evaluation phase, consumer roles—without access, after grant, and after revoke—were dynamically assigned. Meanwhile, the owner role was fixed and assigned to a single user. This fixed assignment ensured consistency in testing the system's functionalities and facilitated a streamlined evaluation process. The scenario focused on assessing the system's access control mechanisms for facial data protection, with particular emphasis on the visibility of facial regions in shared images across different user roles.

For this scenario, a single image owner, Alice, and multiple consumers (e.g., Bob, Charlie, etc.) were defined. The number of consumers varied depending on the number of faces detected in each image. Specifically, in the "Single" category, there was one consumer per image, whereas in the "Small", "Medium", and "Large" categories, the number of consumers corresponded to the number of detected faces in the images within each category. Across all face categories, the total number of consumers was 6607 users, which matched the total number of faces detected across all images. The distribution of consumers across the face categories is presented in Table 4.

Table 4. Consumer distribution across face categories.

Face Category	Total Number of Consumers
Single	996
Small	2127
Medium	1275
Large	2209

In evaluating the system's performance, we employed automated testing to quantify the visibility of facial regions. This was achieved by calculating the average correlation values across horizontal, vertical, and diagonal directions. The image $I(x,y)$ was compared with its shifted versions, including horizontally shifted $I_H(x,y) = I(x,y+1)$, vertically shifted $I_V(x,y) = I(x+1,y)$, and diagonally shifted $I_D(x,y) = I(x+1,y+1)$. The correlation values across these directions were calculated using Equations (1), (2), and (3), respectively.

$$\rho_H = \frac{\sum_{x,y}(I(x,y) - \bar{I})(I_H(x,y) - \bar{I_H})}{\sqrt{\sum_{x,y}(I(x,y) - \bar{I})^2 \sum_{x,y}(I_H(x,y) - \bar{I_H})^2}} \tag{1}$$

$$\rho_V = \frac{\sum_{x,y}(I(x,y) - \bar{I})(I_V(x,y) - \bar{I_V})}{\sqrt{\sum_{x,y}(I(x,y) - \bar{I})^2 \sum_{x,y}(I_V(x,y) - \bar{I_V})^2}} \tag{2}$$

$$\rho_D = \frac{\sum_{x,y}(I(x,y) - \bar{I})(I_D(x,y) - \bar{I_D})}{\sqrt{\sum_{x,y}(I(x,y) - \bar{I})^2 \sum_{x,y}(I_D(x,y) - \bar{I_D})^2}} \tag{3}$$

where $\bar{I}$, $\bar{I_H}$, $\bar{I_V}$, and $\bar{I_D}$ are the mean values of I, I_H, I_V, and I_D, respectively.

Figure 8 compares original and encrypted face image pixel correlations. The original image exhibits strong spatial correlations in the horizontal, vertical, and diagonal directions, as shown by clustered scatter plots. In contrast, the encrypted image shows a uniform distribution, indicating the removal of spatial correlations by the encryption process.

We used the average correlation values across all directions to quantify the visibility of facial regions in shared images, as shown in Equation (4).

$$\rho_{\text{avg}} = \frac{\rho_H + \rho_V + \rho_D}{3} \tag{4}$$

A threshold value of 0.1 was used to determine facial region visibility. If ρ_{avg} was below the threshold, the facial region was considered visible; otherwise, it was considered protected.

Finally, system performance was evaluated by comparing expected and actual results for operations such as face encryption, key splitting, face decryption, and access revocation. Expected results were based on system design, while actual results were obtained through automated testing. This evaluation comprehensively assessed the system's access control mechanism for facial data protection, emphasizing facial region visibility in shared images for different user roles. Results are presented in the following section.

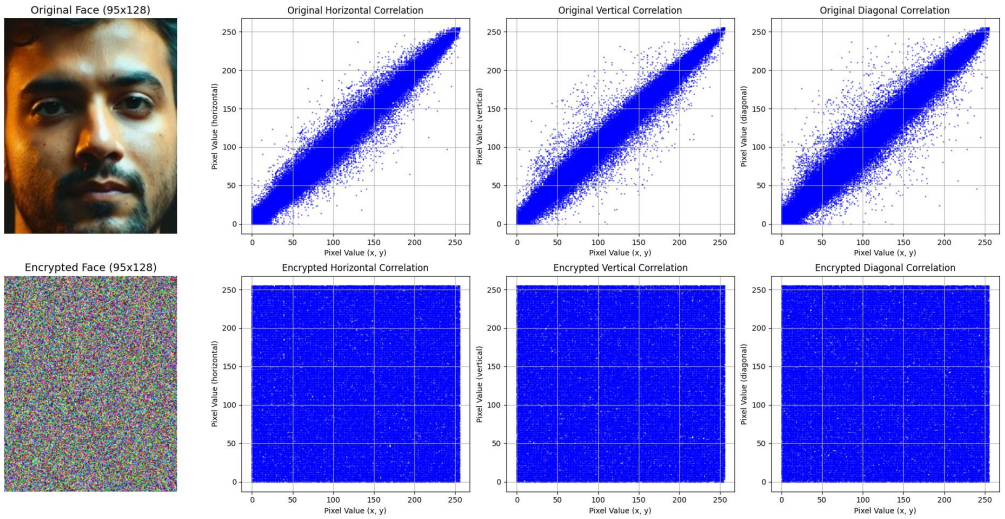


Figure 8. Comparison of original (top row) and encrypted (bottom row) face image pixel correlations.

5.2. System Evaluation

5.2.1. Performance Analysis of Face Encryption Across Group Sizes

We evaluated the system's performance in encrypting facial regions across different group sizes to assess scalability and efficiency. The performance metrics for the CSP component include average detection time, face encryption time, and overall execution time. For the server, we measured average processing time and image storage overhead. The results are presented in Table 5.

Table 5. Performance analysis of face encryption across group sizes.

	CSP			Server	
Face Category	**Avg. Detection Time (ms)**	**Avg. Face Encryption Time (ms)** [1]	**Avg. Overall Execution Time (ms)**	**Avg. Processing Time (ms)**	**Avg. Image Storage Overhead (%)**
Single	198.22 ± 51.62	0.64 ± 0.74	303.82 ± 87.01	340.83 ± 90.98	1245.57 ± 292.26
Small	190.21 ± 18.26	0.48 ± 0.54	289.68 ± 32.45	325.14 ± 37.01	1156.74 ± 206.04
Medium	186.28 ± 14.96	0.44 ± 0.39	281.76 ± 24.64	314.59 ± 28.57	1060.01 ± 170.07
Large	186.44 ± 12.42	0.52 ± 0.22	283.03 ± 26.14	311.96 ± 27.30	1004 ± 123.62

[1] The reported face encryption time represents the average total processing duration for all faces detected in a single image.

The results indicate that the system's performance in encrypting facial regions is consistent across different group sizes. The average detection time ranged from 186.28 ms to 198.22 ms, with the smallest group size (Single) exhibiting the highest detection time. The average face encryption time was consistent across all group sizes, ranging from 0.44 ms to 0.64 ms. The average overall execution time was also consistent, with values ranging from 281.76 ms to 303.82 ms. The average processing time in the server was similar across group sizes, with values ranging from 311.96 ms to 340.83 ms. These results demonstrate the system's scalability and efficiency in encrypting facial regions across different group sizes.

However, the storage requirement for images with encrypted faces is high. As shown in Table 5, the need for lossless storage leads to a substantial increase in file size, typically ranging from 10 to 12 times the size of the original image, which is encoded in JPG format. The average storage overhead was highest for the single group size, at 1245.57%, and lowest for the large group size, at 1004%.

In our work, we attempted to store the output image using JPG encoding. However, this encoding prevented the encrypted image from being decrypted due to compression artifacts. Even at the highest quality setting (100), JPG encoding still resulted in decryption failure. Therefore, the current version of our system uses only PNG encoding to store encrypted images, which consequently leads to high storage overhead.

This limitation highlights a key direction for future research: optimizing encryption and storage strategies to reduce storage overhead while preserving the integrity of encrypted facial regions. Such advancements would be especially valuable for large-scale deployments where storage costs are a significant concern. Our storage evaluation, therefore, not only identifies a practical limitation of the current system but also emphasizes a critical area for further investigation in privacy-preserving image sharing.

5.2.2. Performance Evaluation of Key Split Operations

We evaluated the performance of key-splitting operations in the context of multi-user access to facial data. This evaluation aimed to assess the system's efficiency in granting decryption access to designated consumers. The key-splitting process involves generating key shares and encrypting them, ensuring that only designated consumers are able to access the facial data. The performance metrics included the average CSP split time, average CSP peak memory usage, and key-split success rate. The results are presented in Table 6.

Table 6. The performance of key split operations.

Number of Consumers [1]	Number of Images	Total Key Splits	Avg. CSP Split Time (ms)	Avg. CSP Peak Memory (MB)	Key Split Success Rate (%) [2]
1	996	996	70.39 ± 6.82	4333.06 ± 500.83	100%
2	342	684	69.99 ± 5.4	4921.35 ± 51.68	100%
3	147	441	70.05 ± 7.61	4915.35 ± 52.41	100%
4	128	512	69.21 ± 5.5	4915.68 ± 51.16	100%
5	98	490	69.79 ± 10.96	4927.14 ± 43.45	100%
6–10	171	1275	68.94 ± 5.3	4905.24 ± 70.65	100%
11–65	118	2209	68.91 ± 7.02	4988.1 ± 19.69	100%

[1] The number of consumers refers to the count of detected faces in an image, where each face corresponds to a unique consumer (i.e., the individual depicted in that face). [2] Success rate denotes the percentage of images for which the key-splitting operation successfully granted decryption access to all designated consumers.

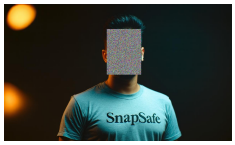
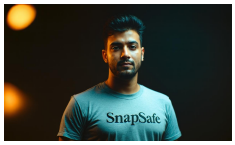
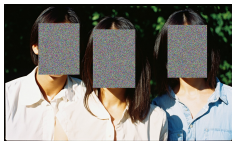
The results indicate that the system efficiently performs key-splitting operations across different group sizes, with consistent performance metrics. The average CSP split time ranged from 68.91 ms to 70.39 ms, with minimal variation across group sizes. The average CSP peak memory usage was also consistent, with values ranging from 4333.06 MB to 4988.1 MB. The key-split success rate was 100% for all group sizes, indicating that the system effectively granted decryption access to designated consumers. These results demonstrate the system's efficiency in managing key-splitting operations for multi-user access to facial data.

5.2.3. Performance Evaluation of Face Decryption Operations

We evaluated the performance of face decryption operations for both the image owner and consumers to assess the system's efficiency in decrypting facial data. For the image owner, the decryption process involves directly retrieving the symmetric key for the image and decrypting the facial regions in the shared images. For consumers, the decryption process requires reconstructing the symmetric key from key shares provided by the owner, followed by decryption of the facial regions using the reconstructed key. The results are

presented in Table 7. Performance metrics include the average decryption time for both the owner and consumers.

Table 7. The Performance of face decryption operations for image owner and consumers.

Encrypted Face Image	Decrypted Face Image	Face Category	Number of Faces	Decryption Time by Owner (ms)	Decryption Time by Consumer (ms)
		Single	1	215.17	211.08
		Small	3	275.75	285.57 ± 15.66
		Medium	8	218.62	239.57 ± 35.62
		Large	11	227.81	253.46 ± 40.87

The decryption time for the owner ranged from 215.17 ms to 275.75 ms, while for consumers it ranged from 211.08 ms to 285.57 ms. These results indicate that the system efficiently decrypts facial data for both groups, with minimal variation in decryption time across different face categories.

The slightly higher decryption time for consumers is attributed to the need for reconstructing the symmetric key from key shares before decrypting the facial regions. However, the minimal difference in decryption time suggests that the system efficiently manages decryption operations for both the image owner and consumers, despite the additional steps required for key reconstruction in the consumer decryption process.

5.2.4. Performance Evaluation of Revocation Operations

We evaluated the performance of revocation operations to assess the system's efficiency in removing decryption access for users previously authorized by the image owner. The revocation process involves deleting key shares associated with revoked users, ensuring they can no longer access the facial data.

The revocation process was conducted in three steps, with the number of revoked users increasing at each step. In the final step, all consumer access was revoked, ensuring that no consumers retained access to the facial data. The performance metrics consist of the average server processing time and the average CSP revocation time. The results are presented in Table 8.

The results indicate that the system efficiently performs revocation operations, with consistent performance metrics across different revocation steps. The average server processing time ranged from 11.95 ms to 15.7 ms, with minimal variation across revocation steps. The average CSP revocation time was also consistent, with values ranging from 5.62 ms to 7.19 ms. These results demonstrate the system's efficiency in managing revo-

cation operations, ensuring that revoked users no longer have access to facial data while maintaining access for authorized users.

Table 8. Performance evaluation of revocation operations.

Initial Authorized Users	Revocation Step	Revoked Users [1]	Remaining Authorized Users [2]	Avg. Server Processing Time (ms)	Avg. CSP Revocation Time
18	1st	5	13	15.7 ± 8.79	5.77 ± 0.52
	2nd	6	7	12.51 ± 0.79	6.02 ± 0.69
	3rd	7	0	12.58 ± 1.35	6.01 ± 0.82
27	1st	8	19	13.77 ± 2.13	7.19 ± 2.11
	2nd	9	10	12.98 ± 3.42	6.63 ± 3.52
	3rd	10	0	12.89 ± 2.09	6.45 ± 1.92
36	1st	11	25	12.27 ± 0.9	5.8 ± 0.59
	2nd	12	13	12.85 ± 2.91	5.69 ± 0.77
	3rd	13	0	11.95 ± 0.72	5.62 ± 0.62

[1] The number of revoked users has been verified to ensure that their access has been successfully removed. [2] The number of remaining authorized users has been validated to confirm their continued access to the face.

6. Conclusions

This paper presents a facial privacy protection system designed as a proactive measure to protect facial data in shared images. The system encrypts facial regions before an image is published, ensuring privacy at the source before others can access it. Building on our previous work on a single-party face protection system, SnapSafe, this study introduces a file-level facial protection mechanism with dynamic multi user access control. The proposed system categorizes users into two roles: image owners, who upload and manage images, and image consumers, who request access to protected facial data. Image owners can grant or revoke access to specific consumers, ensuring controlled facial data sharing through access limitations and an encryption-based permission system.

The system employs key splitting for secure access control, an owner-centric permission model for granting and revoking access, and a request-based mechanism that allows consumers to request facial data access, subject to owner approval.

During the evaluation, the system was tested with varying numbers of faces, categorized into single, small, medium, and large groups. Results demonstrate the system's efficiency in safeguarding facial data while ensuring seamless access in multi-user environments. The encryption of facial data was completed in an average time of less than 350 ms, measured from the request to the response of the server, reflecting real-world performance when network variability is excluded. The system maintained a 100% success rate in key-splitting operations across all group sizes, ensuring secure access sharing. Facial data decryption for both image owners and consumers was averaged under 286 ms, while revocation operations were efficiently processed with average server response times below 16 ms. These results highlight the system's ability to balance security and usability for facial data protection.

For future work, we aim to enhance system scalability by optimizing data storage and exploring encryption schemes that are resilient to lossy compression. This will ensure that encrypted images do not occupy excessive storage while still allowing facial data to be decrypted even after image compression. Additionally, we will develop fine-grained access control that enables image owners to grant selective access to specific facial regions.

These improvements will improve user control over facial data sharing, ensuring that image owners can grant access to specific facial data rather than revealing all faces to authorized users.

Author Contributions: Conceptualization, A.S. and S.H.; methodology, A.S. and S.H.; software, A.S.; validation, A.S.; formal analysis, A.S., S.H., Y.K. and Y.N.; investigation, A.S., S.H., Y.K. and Y.N.; resources, A.S. and Y.N.; data curation, A.S.; writing—original draft preparation, A.S. and S.H.; writing—review and editing, A.S., S.H., Y.K. and Y.N.; visualization, A.S.; supervision, S.H. and Y.N.; project administration, Y.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Acknowledgments: During the preparation of this manuscript, the authors utilized the Stable Diffusion 3.5 Medium AI model to generate test images for evaluating the output and performance of the proposed system. The generated images were used without alteration, except for necessary modifications to accurately reflect the system's output. The authors have reviewed the content and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A. Image Generation for System Evaluation

The appendix provides additional details on the prompts used for image generation with the Stable Diffusion 3.5 Medium model. These prompts, comprising both positive and negative components, were designed to control the number of people in the generated images and specify various attributes. The first and second prompts successfully produced the intended number of people.

However, the third and fourth prompts did not generate the exact number of people as specified due to inherent challenges in synthesizing images of groups with a relatively large number of people. Nevertheless, the resulting images were sufficient for inclusion in the medium and large groups, respectively. Table A1 lists the prompts used to evaluate the system's applicability in real-world scenarios, along with the hyperparameters employed during image generation.

Table A1. Prompts for generating images using Stable Diffusion 3.5 to evaluate the system's applicability in real-world scenarios.

Positive Prompt	Negative Prompt	Hyperparameters	Generated Image
Cinematic photograph of a man wearing a t-shirt with the word 'SnapSafe' printed on it. Captured in a 35 mm film style with a shallow depth of field, rich bokeh, and a natural, professional lighting setup. The image is highly detailed, sharp, and rendered in 4 K resolution, evoking the texture and warmth of analog film.	bad hands, missing fingers, too many fingers, deformed limbs	steps: 85, cfg: 5.5, sampler: DPM++ 2M, scheduler: sgm_uniform, denoise: 1.0	

Table A1. *Cont.*

Positive Prompt	Negative Prompt	Hyperparameters	Generated Image
Wide-angle shot of three persons with a sharp, high-quality lens, full upper body, free from distortions or aberrations. The face is naturally proportioned, with balanced lighting and a realistic depth of field, color negative, with sunlight filtering shadows on their faces, in the style of instant film, KodakT-Max 100, color negative film, add noise, grain	bad hands, missing fingers, too many fingers, deformed limbs	steps: 40, cfg: 5.5, sampler: DPM++ 2M, scheduler: sgm_uniform, denoise 1.0	
The image is a group photo of a group of young men posing for a selfie in a kitchen. There are nine men in the photo.	incorrect anatomy, missing fingers, too many fingers, twins	steps: 65, cfg: 6.5, sampler: DPM++ 2M, scheduler: sgm_uniform, denoise: 1.0	
A high-resolution photograph of a 20 to 25 men wearing casual T-shirts standing in front of “ISEC” sign, close photograph, soft natural daylight, creating a warm, realistic, and slightly nostalgic atmosphere, 0.35 mm lens	bad fingers, extra fingers, asymmetric face, missing legs, extra legs, wrong anatomy, twins	steps: 100, cfg: 10, sampler: DPM++ 2M, scheduler: sgm_uniform, denoise: 1.0	

References

1. Zani, A.A.A.; Norman, A.A.; Ghani, N.A. Motivating Factors to Self-Disclosure on Social Media: A Systematic Mapping. *IEEE Trans. Prof. Commun.* **2022**, *65*, 370–391. [CrossRef]
2. Wen, Y.; Liu, B.; Song, L.; Cao, J.; Xie, R. Facial Recognition Technology and the Privacy Risks. In *Face De-Identification: Safeguarding Identities in the Digital Era*; Springer Nature: Cham, Switzerland, 2024; pp. 15–20. [CrossRef]
3. Yan, W.Q. Biometrics for Surveillance. In *Introduction to Intelligent Surveillance: Surveillance Data Capture, Transmission, and Analytics*; Springer International Publishing: Cham, Switzerland, 2019; pp. 127–153. [CrossRef]
4. Willoughby, A. Biometric Surveillance and the Right to Privacy [Commentary]. *IEEE Technol. Soc. Mag.* **2017**, *36*, 41–45. [CrossRef]
5. Chandler, S. Social Media Proves Itself to Be the Perfect Tool for Stalkers. Online Article, Forbes. 2019. Available online: https://www.forbes.com/sites/simonchandler/2019/10/11/social-media-proves-itself-to-be-the-perfect-tool-for-stalkers/ (accessed on 21 January 2025).
6. Karras, T.; Laine, S.; Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 4217–4228. [CrossRef] [PubMed]
7. Chen, R.; Chen, X.; Ni, B.; Ge, Y. SimSwap: An Efficient Framework For High Fidelity Face Swapping. In Proceedings of the 28th ACM International Conference on Multimedia, New York, NY, USA, 12–16 October 2020; MM ’20, pp. 2003–2011.
8. Xu, S.; Chen, G.; Guo, Y.X.; Yang, J.; Li, C.; Zang, Z.; Zhang, Y.; Tong, X.; Guo, B. VASA-1: Lifelike Audio-Driven Talking Faces Generated in Real Time. *arXiv* **2024**, arXiv:2404.10667.

9. Rössler, A.; Cozzolino, D.; Verdoliva, L.; Riess, C.; Thies, J.; Niessner, M. FaceForensics++: Learning to Detect Manipulated Facial Images. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–11.
10. Cozzolino, D.; Rössler, A.; Thies, J.; Nießner, M.; Verdoliva, L. ID-Reveal: Identity-aware DeepFake Video Detection. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 15088–15097.
11. Wang, S.Y.; Wang, O.; Zhang, R.; Owens, A.; Efros, A.A. CNN-Generated Images Are Surprisingly Easy to Spot... for Now. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 19 June 2020; pp. 8692–8701. [CrossRef]
12. Hugl, U. Reviewing person's value of privacy of online social networking. *Internet Res.* **2011**, *21*, 384–407. [CrossRef]
13. Liu, C.; Zhu, T.; Zhang, J.; Zhou, W. Privacy Intelligence: A Survey on Image Privacy in Online Social Networks. *ACM Comput. Surv.* **2022**, *55*, 1–35. [CrossRef]
14. Liang, K.; Liu, J.K.; Lu, R.; Wong, D.S. Privacy Concerns for Photo Sharing in Online Social Networks. *IEEE Internet Comput.* **2015**, *19*, 58–63. [CrossRef]
15. Tajik, K.; Gunasekaran, A.; Dutta, R.; Ellis, B.; Bobba, R.B.; Rosulek, M.; Wright, C.V.; Chi Feng, W. Balancing Image Privacy and Usability with Thumbnail-Preserving Encryption. In Proceedings of the Network and Distributed System Security (NDSS) Symposium 2019, San Diego, CA, USA, 24–27 February 2019; pp. 1–15. . [CrossRef]
16. Amon, M.J.; Hasan, R.; Hugenberg, K.; Bertenthal, B.I.; Kapadia, A. Influencing Photo Sharing Decisions on Social Media: A Case of Paradoxical Findings. In Proceedings of the 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 18–21 May 2020; pp. 1350–1366. [CrossRef]
17. Santoso, A.; Huda, S.; Nguyen, T.T.; Kodera, Y.; Nogami, Y. SnapSafe: Enabling Selective Image Privacy Through YOLO and AES-Protected Facial Encryption with QR Code. In Proceedings of the 2024 International Technical Conference on Circuits/Systems, Computers, and Communications (ITC-CSCC), Okinawa, Japan, 2–5 July 2024; pp. 1–5. [CrossRef]
18. Jocher, G.; Chaurasia, A.; Qiu, J. Ultralytics YOLOv8, version 8.0.0, 2023. Available online: https://docs.ultralytics.com/models/yolov8 (accessed on 17 December 2024).
19. U.S. Copyright Office. Copyright Registration of Photographs (Circular 42). 2021. Available online: https://www.copyright.gov/circs/circ42.pdf (accessed on 24 February 2025).
20. European Parliament and Council. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons with Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC (General Data Protection Regulation). Official Journal of the European Union, L 119, 4 May 2016; pp. 1–88. Available online: https://eur-lex.europa.eu/eli/reg/2016/679/oj (accessed on 24 February 2025).
21. You, X.; Zhao, X.; Wang, Y.; Sun, W. Generation of Face Privacy-Protected Images Based on the Diffusion Model. *Entropy* **2024**, *26*, 479. [CrossRef] [PubMed]
22. He, X.; Zhu, M.; Chen, D.; Wang, N.; Gao, X. Diff-Privacy: Diffusion-Based Face Privacy Protection. *IEEE Trans. Circuits Syst. Video Technol.* **2024**, *34*, 13164–13176. [CrossRef]
23. Suwała, A.; Wójcik, B.; Proszewska, M.; Tabor, J.; Spurek, P.; Śmieja, M. Face Identity-Aware Disentanglement in StyleGAN. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–8 January 2024; pp. 5222–5231.
24. Yang, Y.; Lyu, J.; Wang, R.; Wen, Q.; Zhao, L.; Chen, W.; Bi, S.; Meng, J.; Mao, K.; Xiao, Y.; et al. A digital mask to safeguard patient privacy. *Nat. Med.* **2022**, *28*, 1883–1892. [CrossRef] [PubMed]
25. Xu, K.; Guo, Y.; Guo, L.; Fang, Y.; Li, X. My Privacy My Decision: Control of Photo Sharing on Online Social Networks. *IEEE Trans. Dependable Secur. Comput.* **2017**, *14*, 199–210. [CrossRef]
26. Tang, L.; Ma, W.; Grobler, M.; Meng, W.; Wang, Y.; Wen, S. Faces are Protected as Privacy: An Automatic Tagging Framework Against Unpermitted Photo Sharing in Social Media. *IEEE Access* **2019**, *7*, 75556–75567. [CrossRef]
27. Jocher, G.; Chaurasia, A.; Stoken, A.; Borovec, J.; NanoCode012; Kwon, Y.; Michael, K.; TaoXie; Fang, J.; imyhxy; et al. YOLOv5 SOTA Realtime Instance Segmentation, v7.0, Zenodo, 2022. Available online: https://doi.org/10.5281/zenodo.7347926 (accessed on 17 December 2024).
28. Geng, S.; Li, J.; Zhang, X.; Wang, Y. An Image Encryption Algorithm Based on Improved Hilbert Curve Scrambling and Dynamic DNA Coding. *Entropy* **2023**, *25*, 1178. [CrossRef] [PubMed]
29. Mfungo, D.E.; Fu, X. Fractal-Based Hybrid Cryptosystem: Enhancing Image Encryption with RSA, Homomorphic Encryption, and Chaotic Maps. *Entropy* **2023**, *25*, 1478. [CrossRef] [PubMed]
30. Zhang, Y.; Zeng, J.; Yan, W.; Ding, Q. RBFNN-PSO Intelligent Synchronisation Method for Sprott B Chaotic Systems with External Noise and Its Application in an Image Encryption System. *Entropy* **2024**, *26*, 855. [CrossRef] [PubMed]
31. HashiCorp. Vault v1.18.x Documentation. Available online: https://developer.hashicorp.com/vault/docs/v1.18.x (accessed on 14 December 2024).

32. HashiCorp. Auto-auth Method: Application Roles (AppRole). Available online: https://developer.hashicorp.com/vault/docs/v1.18.x/agent-and-proxy/autoauth/methods/approle (accessed on 14 December 2024).
33. Docker. Docker Engine Release Notes: Version 27. Available online: https://docs.docker.com/engine/release-notes/27/ (accessed on 24 November 2024).
34. Yang, S.; Luo, P.; Loy, C.C.; Tang, X. WIDER FACE: A Face Detection Benchmark. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 5525–5533.
35. Ultralytics. YOLOv8m Pre-Trained Model. 2023. Available online: https://github.com/ultralytics/assets/releases/download/v8.2.0/yolov8m.pt (accessed on 15 November 2024).
36. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. In *Proceedings of the Computer Vision—ECCV 2014, Zurich, Switzerland, 6–12 September 2014*; Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T., Eds.; Springer: Cham, Switzerland, 2014; pp. 740–755.
37. Ultralytics. YOLOv8 Performance Metrics. 2023. Available online: https://docs.ultralytics.com/models/yolov8/#performance-metrics (accessed on 17 December 2024).
38. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. *arXiv* **2019**, arXiv:1912.01703. [CrossRef]
39. Stability AI. Stable Diffusion 3.5 Medium. 2024. Available online: https://huggingface.co/stabilityai/stable-diffusion-3.5-medium (accessed on 21 February 2025).