Balancing Image Privacy and Usability with Thumbnail-Preserving Encryption

Kimia Tajik*, Akshith Gunasekaran*, Rhea Dutta†§, Brandon Ellis*, Rakesh B. Bobba*, Mike Rosulek*, Charles V. Wright‡ and Wu-chi Feng‡
*Oregon State University, †Cornell University, ‡Portland State University
{tajikk, gunaseka, ellibran, rakesh.bobba, rosulekm}@oregonstate.edu, rd434@cornell.edu, {cvwright, wuchi}@cs.pdx.edu

***Abstract*—In this paper, we motivate the need for image encryption techniques that preserve certain visual features in images and hide all other information, to balance privacy and usability in the context of cloud-based image storage services. In particular, we introduce the concept of ideal or exact Thumbnail-Preserving Encryption (TPE), a special case of format-preserving encryption, and present a concrete construction. In TPE, a ciphertext is itself an image that has the same thumbnail as the plaintext (unecrypted) image, but that provably leaks nothing about the plaintext beyond its thumbnail. We provide a formal security analysis for the construction, and a prototype implementation to demonstrate compatibility with existing services. We also study the ability of users to distinguish between thumbnail images preserved by TPE. Our findings indicate that TPE is an efficient and promising approach to balance usability and privacy concerns for images. Our code and a demo are available at: *http://photoencryption.org*.**

I. Introduction

The advent of affordable high-resolution digital cameras has allowed us to capture many snapshots of our daily lives. In particular, cameras on mobile phones and other smart, hand-held devices have made it extremely easy to capture everyday activities, from encounters with police and political protests, to vacation time with friends and family, and even the most intimate moments of life. Given the ubiquitous access to fast mobile networks, a vast number of digital images and videos that are recorded are transmitted or stored with third-party storage (service) providers in the cloud. For example, Apple, Google, Microsoft, and Dropbox all offer services to automatically synchronize photos from mobile devices to their clouds. Users of social networks share more than one billion new photos each week [2]. While these third-party service providers potentially enable users to better store, share, organize, and manage their images and videos, the centralization poses a serious threat to their privacy. Data breaches are becoming more common and attackers have recently gained access to thousands of accounts on the most popular services [11], [32]. CNN has reported on a breach to a photo-sharing site that exposed many users in 2012[1]. In other cases, cloud services themselves have exploited user data for their own benefit [35] or to satisfy a secret request from a third party [4].

Fig. 1: Left: the original image with its thumbnail. Middle: the TPE-encrypted image with its thumbnail. Right: traditionally encrypted image with its thumbnail.

Encrypting the images before uploading to the cloud services alleviates the privacy concerns as the service providers would only have access to the ciphertext. However, a downside of such a solution is that it undercuts the convenience provided by such services. Users can no longer browse, organize, and manage their images on the cloud side because they would be unable to distinguish between the images in ciphertext form (See rightmost image in Figure 1). Even if users concede to images being available in a decrypted form while they have an active in-person session with the cloud server, this would most likely require service providers to be willing to fundamentally modify their services to specifically support encrypted images. For example, image tagging and labeling approaches may allow private browsing of encrypted images when combined with searchable encryption, as demonstrated by Pixek App[2]. However, this would require the service provider to modify their service to support a specific searchable encryption scheme, not to mention additional effort on the part of the user to tag/classify their images. We elaborate on this and other approaches in Section III.

Proposed Approach: We propose the ideal Thumbnail-Preserving Encryption (TPE) method as a solution for balancing image privacy and usability, particularly for the cloud-based image storage scenarios. TPE, as the name indicates, is a special case of format-preserving encryption scheme [3] that preserves (only) the thumbnail of the original image. That is, the ciphertext is also an image whose thumbnail (at some specified resolution) is the same as that of the original plaintext image (See middle image in Figure 1).

[1]CNN: Photobucket breach floods Web with racy images. http://articles.cnn.com/2012-08-09/tech/tech_photobucket-privacy-breach.

[2]www.wired.com/story/pixek-app-encrypts-photos-from-camera-to-cloud/

§Rhea Dutta worked on this paper while interning at Oregon State University

Network and Distributed Systems Security (NDSS) Symposium 2019
24-27 February 2019, San Diego, CA, USA
ISBN 1-891562-55-X
https://dx.doi.org/10.14722/ndss.2019.23432
www.ndss-symposium.org

Suppose a user encrypts images under TPE and uploads the ciphertexts to a cloud service. The service will treat these encrypted images as any other image, providing an interface to browse/manage them based on their thumbnails. Importantly, standard thumbnails of TPE-encrypted images are human-recognizable (as low-resolution versions of the plaintext images). Hence, the user gets the benefit of encryption, while still being able to manage and browse their images **without any modification to the cloud service backend** or the user's familiar usage pattern. The cloud service itself sees only TPE-encrypted images and therefore learns no more than what can be inferred from a low-resolution thumbnail of each image. In other words, the information leakage is quantified exactly in terms of the preserved thumbnail. High-level qualitative interpretation of that leakage (w.r.t. privacy) is however, highly content and context-dependent as will be evident from the discussion later in the paper (Sections II and IX).

Our hypothesis is that, by controlling the resolution of the thumbnail preserved by the ciphertext, users can achieve a good balance between usability and privacy. For instance, Figure 2 shows TPE-encrypted images with different block sizes along with their corresponding preserved thumbnails. Through a qualitative user study, we show that users are able to successfully distinguish between and identify TPE-encrypted images using thumbnails with low enough resolutions that even a state-of-the-art recognition algorithm fails to recognize the images (Section VII).

In particular, our approach exploits the human ability to recognize images and especially faces even when they are highly distorted when users *know* or *have seen* them before [12]. This ability is known to be stronger if users themselves created or captured the image [20]. This ability has been leveraged by other works in defending against shoulder surfing attacks in image-based authentication systems (*e.g.,* [14], [16]) and image browsing [40].

An approximate-TPE scheme has been previously proposed by Wright *et al.* [41]. The scheme is approximate in the sense that ciphertext images leak more information than just the thumbnail, and the encryption-decryption process is somewhat lossy (in particular, washing out colors). In a follow-up work [25], approximate-TPE schemes that preserve a perceptually close but not the exact thumbnail have been proposed. Those schemes also tend to impact the quality of decrypted images in some cases. In contrast, in this work, we propose an **ideal**-TPE scheme (Section V) that preserves the exact thumbnail and is lossless. We evaluate the ideal-TPE scheme from security (Section V-B), performance (Section VI) and user experience (Section VII) perspectives.

Contributions: The main contributions of this work are:

- We formalize the technical requirements for Ideal-Thumbnail-Preserving Encryption (Ideal-TPE) as a special case of format-preserving encryption (FPE), and provide a discussion of its general feasibility (Section IV).
- We provide a concrete construction for an Ideal-TPE scheme with a formal cryptographic proof of security (Section V).
- We show that Ideal-TPE is compatible with existing services by implementing a proof-of-concept browser plug-in to work with Google Photos (Section VI).
- We provide evidence for the usability of TPE-encrypted images through a qualitative user study where we demonstrate that users can distinguish and identify images using low resolution thumbnails, even when the resolution is low enough that standard computer vision systems fail (Section VII).

II. Threat Model and Scope

Privacy threats to users' images come from different sources and in different forms. For example, unauthorized access could be gained by curious (or malicious) insiders at the image storage service provider, by hackers who breached the storage provider, or en route to the network. The consequences of privacy compromise depend on the contents of the image and vary greatly. In some cases, the existence of association between two subjects in the image (*e.g.,* photo with a known criminal or disgraced person) could be sensitive information. In others, the activity or content depicted by the image (*e.g.,* leaked racy images) could be privacy sensitive. In the former example, the *recognition or identification* of subjects in the image is implicit: in the latter case, it may or may not be necessary for the subject to be identified. In yet other cases, the image or a version of it is itself publicly available but the identity of the subjects in the image is considered private information (*e.g.,* tagging of public images).

In all the cases discussed, the threat source could be humans or machine learning (ML)-based image analysis algorithms. In this paper, we focus on the scenario of image storage in the cloud where a cloud service provider offers a service for storing and managing photo albums. In this scenario, users face threats from rogue employees of the provider, from third parties who gained access to the provider's network and/or servers, and from the provider itself who may value the data mining opportunity. We aim to protect end users against the recognition threat from humans and machine learning (ML) algorithms in this scenario. Further, we aim to minimize the impact of exposure even in cases where identification is not the main threat.

Given that some information is allowed to leak in TPE for usability, there will be scenarios where there is privacy loss. In other words, there are cases where a thumbnail itself may be sensitive/incriminating or other side-channels might add to the leakage. For example, a close family member might be able to recognize a person in a TPE-protected image and might even be able to deduce the activity even if finer details might be withheld. Similarly, a celebrity might be recognized and their the activity might be inferred in a TPE-protected image, even if the finer details are withheld. On the other hand, there are also many scenarios where the only sensitive information is contained in finer details of an image. For instance, TPE-image might reveal the presence of a car but make its license plate unreadable, or reveal that there is intimacy while hiding finer details. One could think of the privacy afforded in these scenarios as comparable to that of a (digital) *privacy glass*.

III. Potential Solutions

In this section, two potential solutions for balancing user convenience and privacy using conventional cryptography are addressed. Both of these solutions have shortcomings as discussed in the following.

Fig. 2: TPE-encrypted images with different block sizes (original, 5×5, 10×10, 20×20), and corresponding preserved thumbnails.

Embedding the Thumbnail: One approach to improving usability of encrypted images is to associate a plaintext thumbnail with a ciphertext image. In this instance, the associated thumbnail would be used to preview the encrypted image uploaded to the cloud, while the actual image would be encrypted and thus prevent the leakage of additional information about the image. In fact, JPEG File Interchange Format (JFIF)[3] and Exchangeable Image File Format (Exif)[4] formats for exchanging JPEG-encoded files, provide an option to embed a thumbnail along with the JPEG-encoded image.

This approach, however, suffers from a practical shortcoming. We tested several popular cloud storage services, including Google Drive, Dropbox, and iCloud, and found that they do not use the embedded thumbnail in their preview mode. Thus, this approach is currently not compatible with existing popular cloud storage services, necessitating the co-operation of storage providers and changes to their software to be useful.

Image Tagging: Another potential solution is to securely associate descriptive tags to images before encrypting and storing them in the cloud. The tags can be stored either locally or in the cloud encrypted. This can be used to distinguish between images and to find desired images.[5] The problem with this approach is the effort it requires from the user's side. Before uploading images to the cloud, a user has to tag every image and has be careful to tag them in a way that is distinctive and informative enough for the image to be distinguished and retrieved using the tag only. That is, for users to access their uploaded images and select one (or more) among them, they need to read the tags and choose the images that are of interest to them, which requires effort as well.

Even if the tagging is automated using computer vision algorithms, browsing images using tags may not come naturally to many users. Also, even if this effort could further be reduced by enabling search function on the tags, conducting a search using tags requires users to have a convention for tagging images, and they need to remember the tags over potentially long periods to be able to search for them later. Even then, this approach is only reasonable when users are looking for a specific image and is not suitable when users want to browse images without a specific image in mind (*e.g.*, looking for a good family picture to put on a Christmas card).

[3]JPEG File Interchange Format, Version 1.02, September 1992. https://www.w3.org/Graphics/JPEG/jfif3.pdf

[4]Exchangeable Image File Format for Digital Still Cameras, Exif Version 2.2, April 2002. http://www.exif.org/Exif2-2.PDF

[5]An App that encrypt your photos from camera to the cloud. https://www.wired.com/story/pixek-app-encrypts-photos-from-camera-to-cloud?mbid=social_twitter

IV. TPE Definition, Framework, and Construction

A. Definitions

1) Nonce-based encryption: Thumbnail-preserving encryption is a specific type of nonce-based encryption. It is known that probabilistic encryption requires ciphertext expansion in order to achieve security against chosen plaintext attacks. In our setting, we want the plaintext image and ciphertext image to have the same dimensions, etc. For compatibility with existing services, we cannot require the service provider to store any new encryption-specific auxiliary data outside of the image itself. These goals preclude any ciphertext expansion. In practice, the image filename, date, or other commonly preserved image metadata can serve as a nonce.

$\underline{\mathsf{Enc}_K(T, M) \to C}$: The (deterministic) encryption algorithm takes a symmetric key $K \in \{0,1\}^\lambda$, a nonce $T \in \{0,1\}^*$, a plaintext M, and returns a ciphertext C

$\underline{\mathsf{Dec}_K(T, C) \to M}$: The decryption algorithm takes a symmetric key K, a nonce T, a ciphertext C, and returns a plaintext M

2) Format-preserving encryption: Thumbnail-preserving encryption is a special case of format-preserving encryption (FPE) [3], specifically unique-format FPE. We now give definitions for unique-format FPE, which are different in syntax but equivalent to the ones of Bellare *et al.* [3].

Let $\mathcal{M}$ be the set of plaintexts and let Φ be any function defined on $\mathcal{M}$ that represents the *property* we wish to preserve. Looking ahead, thumbnail-preserving corresponds to choosing $\mathcal{M}$ to be the set of images (in some encoding), and $\Phi(M)$ to be dimensions of image M and its low-resolution thumbnail.

Definition 1: An encryption scheme is **Φ-preserving** (over $\mathcal{M}$) if it satisfies the following properties for all K, T, M:

(Decryption is correct:)	$\mathsf{Dec}_K(T, \mathsf{Enc}_K(T, M)) = M$
(Format-preserving:)	$\mathsf{Enc}_K(T, M) \in \mathcal{M}$
(Ciphertexts preserve Φ:)	$\Phi(\mathsf{Enc}_K(T, M)) = \Phi(M)$

The definition is equivalent to the original FPE definition (for unique-formats) by considering $\{C \mid \Phi(C) = \Phi(M)\}$ to be the "slices" of the plaintext space – each item in the

plaintext space clearly belongs to exactly one slice of the format space.

3) Security: Bellare *et al.* define several notions of security for FPE. Below, we present one of their definitions:

Definition 2 ([3]): Let $\mathcal{F}_\Phi$ denote the set of functions $F : \{0,1\}^* \times \mathcal{M} \to \mathcal{M}$ that are "Φ-preserving" in the sense that $\Phi(F(T,M)) = \Phi(M)$, for all T, M.

An FPE scheme has **PRP security** if, for all PPT oracle machines $\mathcal{A}$,

$$\left| \Pr_{K \leftarrow \{0,1\}^\lambda} \left[\mathcal{A}^{\mathsf{Enc}_K(\cdot,\cdot)}(\lambda) = 1 \right] - \Pr_{F \leftarrow \mathcal{F}_\Phi} \left[\mathcal{A}^{F(\cdot,\cdot)}(\lambda) = 1 \right] \right|$$

is negligible in λ.

If the distinguisher is also allowed oracle access to Dec_K, then the notion corresponds to that of a *strong* PRP and is analogous to a CCA-security requirement for encryption. We believe that a weaker definition is also sufficient in the case of thumbnail-preserving encryption.

Definition 3: Let $\mathcal{A}$ be a PPT oracle machine, where its oracle takes two arguments. $\mathcal{A}$ is called **nonce-respecting** if it never makes two oracle calls with the same first argument.

An FPE scheme has **nonce-respecting (NR)** security if it satisfies *Definition 2* but only with respect to nonce-respecting distinguishers.

Recall that in our motivating example usage of TPE, each image has a natural and unique identifier that can be used as a nonce. Under the guarantee that no two images are encrypted under the same nonce, NR security gives the same guarantee as PRP security – namely, that ciphertext images are indistinguishable from randomly chosen images that share the same thumbnail as the plaintext.

4) Thumbnail-preserving encryption: As a general term, **Thumbnail-Preserving Encryption (TPE)** refers to the case where $\mathcal{M}$ is a space of images, and $\Phi(M)$ consists of the dimensions of M along with some kind of low-resolution version of M. In other words, ciphertexts are themselves images with the same dimensions and "thumbnail" as the plaintext, but leak nothing about the plaintext beyond this thumbnail.

We refer to the following parameters as b**-ideal TPE**. For simplicity, $\mathcal{M}$ consists of images whose spatial dimensions are each a multiple of b. $\Phi(M)$ includes the dimensions of M, and, for each $b \times b$ image block, $\Phi(M)$ includes the average (equivalently, the sum) of pixel intensities in that block.

In some cases, it may make sense to slightly relax the requirements of ideal TPE. In [25], Marohn *et al.* give constructions of TPE that leak slightly more than an ideal thumbnail, and/or do not have perfect correctness (*i.e.*, the ciphertext may not preserve the thumbnail exactly, and the decrypted image is only perceptually similar to the original ciphertext). In this work, we focus only on achieving ideal TPE, since it leaks a minimal amount that is easiest to understand.

B. Feasibility of Ideal NR-TPE for Raw Images

We consider images of the following type:

- An image consists of one or more *channels* (*e.g.*, grayscale, RGB, RGB+opacity, YUV, HSV).
- Each channel is a two-dimensional array of pixels, with values/intensities from $\{0, \ldots, d-1\}$ for some d (often, $d = 256$).

An image thumbnail is generated by first dividing the image into $b \times b$ blocks, and computing the average of pixel intensities within that block. The resulting value becomes a single pixel in the thumbnail image.[6]

1) Sum-preserving encryption: Computing an image thumbnail operates independently on each channel of the image and for each $b \times b$ block. Indeed, if one can encrypt a single channel of a single $b \times b$ block in a way that preserves its sum, then one can easily perform thumbnail-preserving encryption. We formalize the operation of a thumbnail-preserving encryption scheme on a single block+channel in the following primitive:

Definition 4: A **sum-preserving encryption** scheme is a scheme with message space $\mathcal{M} = (\mathbb{Z}_d)^n$ and is Φ_{sum}-preserving with respect to $\Phi_{\mathsf{sum}}(v_1, \ldots, v_n) = \sum_i v_i$, where the sum is over the integers.

The main challenge of designing such a scheme comes from the fact that individual elements of the vector are bounded ($\mathbb{Z}_d$) while the sum being preserved is over the integers. If we only wished to preserve the sum mod d, then such an NR-secure scheme would be incredibly easy: simply use a pseudorandom function (with key K and argument T) to choose a random vector whose sum-mod-d is 0, then add this vector to the plaintext.

Feasibility: A sum-preserving encryption scheme can be constructed in a general way using the rank-encipher approach of [3] which we briefly summarize. The purpose of this section is to demonstrate the *feasibility* of the concept, since the construction presented here would not be particularly fast.

With $\mathcal{M} = (\mathbb{Z}_d)^n$ and $s \in \mathbb{Z}$, define

$$\Phi^{-1}(s) = \{\vec{v} \in \mathcal{M} \mid \sum_i v_i = s\}.$$

Let $N_s = |\Phi^{-1}(s)|$ and let $\mathsf{rank}_s : \Phi^{-1}(s) \to \mathbb{Z}_{N_s}$ be a bijection called a **ranking function**.

The basic idea to encrypt a vector $\vec{v}$ is to first compute its sum $s = \sum_i v$, then compute its rank $\mathsf{rank}_s(\vec{v})$. The rank is a number in $\mathbb{Z}_{N_s}$ and can be enciphered by any scheme with this domain. In the case of NR-security, this enciphering step can be a one-time pad (addition mod N_s) by a pseudorandom pad derived from a PRF (with key K and argument T). Then the enciphered rank can be transformed into a vector of integers by unranking via rank_s^{-1}.

Bellare *et al.* [3] show that efficient ranking/unranking is possible for any *regular language*. In this case, one can represent elements in $\Phi^{-1}(s)$ as binary strings using the bars+stars methodology: Elements of $\Phi^{-1}(s)$ can be encoded as strings of 0s and 1s where:

[6] In this idealized setting, the average of a $b \times b$ block is not quantized to $\mathbb{Z}_d$ but is rather a rational (with denominator b^2).

- The total number of 1s is exactly s (1s represent vector components written in unary).
- The total number of 0s is exactly $n-1$ (0s represent boundaries between vector components).
- There are no more than $d-1$ 1s between any two 0s (each vector component is at most $d-1$).

This language is regular and can be represented by a DFA with $O(d^2n^2)$ states. The procedure suggested in [3] results in a ranking/unranking scheme whose computational complexity is $O(d^3n^3)$.

2) Extending sum-preservation to thumbnail-preservation: A sum-preserving encryption can be extended to an ideal thumbnail-preserving scheme in a natural way. Simply encrypt every thumbnail block of every channel independently (in a sum-preserving way) with distinct nonces.

We use the following notation for an image M:

- $(c, d, w, h) \leftarrow \mathsf{Params}(M)$ means that image M has c channels, pixel values in $\mathbb{Z}_d$, width w, and height h.
- $M_b[k, i, j]$ denotes the entire (i, j)th $b \times b$ sub-block of the kth channel, as a vector of length b^2.

$\mathsf{Enc}_K(T, M)$:
  $(c, d, w, h) \leftarrow \mathsf{Params}(M)$, where $w = w'b$ and $h = h'b$
  for $k \in \{1, \ldots, c\}, i \in \{1, \ldots, w'\}, j \in \{1, \ldots, h'\}$:
    $C_b[k, i, j] \leftarrow \mathsf{SPEnc}_K(T \| k \| i \| j, M_b[k, i, j])$
  return C

$\mathsf{Dec}_K(T, C)$:
  $(c, d, w, h) \leftarrow \mathsf{Params}(M)$, where $w = w'b$ and $h = h'b$
  for $k \in \{1, \ldots, c\}, i \in \{1, \ldots, w'\}, j \in \{1, \ldots, h'\}$:
    $M_b[k, i, j] \leftarrow \mathsf{SPDec}_K(T \| k \| i \| j, C_b[k, i, j])$
  return M

Fig. 3: Construction to extend sum-preserving encryption $(\mathsf{SPEnc}, \mathsf{SPDec})$ to $b \times b$ thumbnail-preserving encryption $(\mathsf{Enc}, \mathsf{Dec})$.

In Figure 3, we show the construction of a thumbnail-preserving scheme from a sum-preserving scheme.

Lemma 1: If $(\mathsf{SPEnc}, \mathsf{SPDec})$ is an NR-secure scheme (preserving sum of vector components) then $(\mathsf{Enc}, \mathsf{Dec})$ (Figure 3) is an NR-secure ideal-TPE.

Proof: The proof is straight-forward. We consider an adversary $\mathcal{A}$ with oracle access to $\mathsf{Enc}_K(\cdot, \cdot)$. The view of such an adversary can be exactly simulated by a suitable adversary $\mathcal{A}'$ with oracle access to $\mathsf{SPEnc}_K(\cdot, \cdot)$. If $\mathcal{A}$ never repeats a nonce, then neither does $\mathcal{A}'$. The NR-security of SPEnc is that responses of the $\mathsf{SPEnc}_K(\cdot, \cdot)$ can be replaced by random vectors with the same sum as the input vector. $\mathcal{A}'$ reassembles the responses of SPEnc into a response for $\mathcal{A}$. For a given call to Enc, the response will be uniformly distributed subject to having the same thumbnail as the query image. ∎

V. A Practical TPE Construction

In Section IV-B, we show the feasibility of ideal NR-TPE for raw images, based on the rank-encipher methodology of [3]. However, encrypting a single $b \times b$ image block (*i.e.*, a vector of $n = b^2$ integers) requires $O(d^3b^6)$ complexity, making the construction impractical for any reasonable application on real-world images.

On the other hand, the problem of sum-preserving encryption with $n = 2$ (corresponding to a block of only 2 pixels) is extremely simple and practical. With a vector $(a, b) \in \{0, \ldots, d-1\}^2$ with sum $s = a + b$, the ranking and unranking functions are:

$$\mathsf{rank}_s(a, b) = \begin{cases} a & \text{if } s < d \\ d - a & \text{otherwise} \end{cases}$$

$$\mathsf{rank}_s^{-1}(r) = \begin{cases} (r, s - r) & \text{if } s < d \\ (d - r, s - d + r) & \text{otherwise} \end{cases}$$

We propose a TPE approach (more precisely, an approach for sum-preserving encryption) that reduces the problem to the simple $n = 2$ case, using ideas inspired by substitution-permutation ciphers.

A. Description of Scheme

Our sum-preserving encryption scheme is described in Figure 4 and is based on pixel-level substitutions and permutations. The main idea is to alternate between two basic operations:

- **Substitution:** all pixels/integers are grouped into pairs. A simple sum-preserving encryption is applied, in parallel, to each *pair* (*e.g.*, using the rank-encipher approach described previously). Each such substitution step is done with a distinct nonce.
- **Permutation:** *all* pixels/integers in the block/vector are permuted with a random permutation derived from a PRF. More precisely, we sample a permutation π over $\{1, \ldots, n\}$ and shuffle the pixels/integers according to π. In practice, this can be implemented via the Fisher-Yates shuffle algorithm, with random choices obtained from PRF.

It is easy to see that each step indeed preserves the sum of pixels/integers. This alternating substitution-permutation structure is repeated for some R number of rounds. In the following sections, we discuss the security of this construction, specifically regarding the choice of R. Decryption works by performing the same steps in reverse order. Importantly, each substitution step and permutation step is reversible.

B. Markov Chain Analysis

To analyze the security of our scheme, we model the encryption algorithm as a Markov chain and relate its security to the mixing time of that Markov chain.

Definition 5: A **finite Markov chain** is a process that moves among the elements of a finite set Ω based on a transition matrix called P, such that at $x \in \Omega$, the next state is chosen according to a fixed probability distribution $P(x, \cdot)$. The x_{th} row of P is the distribution $P(x, \cdot)$. P is stochastic, that is, its entries are all non-negative and the sum of entries in each row equals one [23].

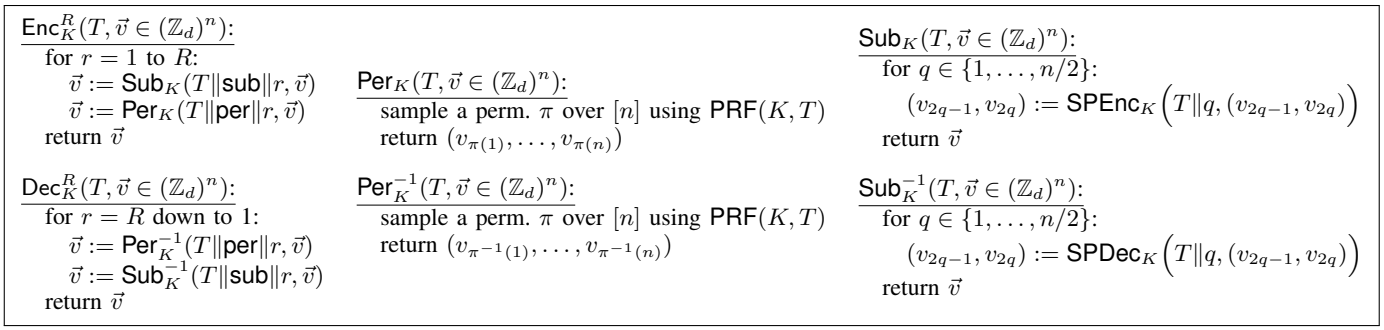

Fig. 4: Sum-preserving encryption scheme for $(\mathbb{Z}_d)^n$, based on substitution-permutation networks. PRF refers to a pseudorandom function, and $(\mathsf{SPEnc}, \mathsf{SPDec})$ refers to a sum-preserving encryption scheme for $n = 2$. R denotes the number of rounds.

Definition 6: Suppose there is a distribution π over Ω satisfying $\pi = \pi P$. Then π is a **stationary distribution** of the Markov chain [23].

It is known that the distribution over Markov chain states approaches a stationary distribution after sufficient rounds.

The scheme as a Markov chain. For each sum s, $\Phi(s)$ denotes the set of vectors in $(\mathbb{Z}_d)^n$ whose sum is s. These vectors correspond to the states of the Markov chain. The transition probability matrix (P) of the Markov chain represents the probability of transitioning from one state to another by means of one substitution plus one permutation (*i.e.,* one encryption round). In our scheme, such transition probabilities are determined by a pseudorandom function. Our conceptual Markov chain instead models the probabilities in an ideal manner. We consider the permutation round to permute values in a perfectly uniform manner, and each substitution (applied to 2 pixel values) to be uniformly chosen. Of course, the security of the underlying PRF is that it induces probabilities that are indistinguishable from this ideal Markov chain (we elaborate below).

In *Lemma 9* in the Appendix, we prove that for the Markov chain model of our TPE scheme, the uniform distribution on $\Phi(s)$ is the unique stationary distribution, therefore, the chain (hence the encryption process) converges to this distribution.

C. Mixing Time and Security

The mixing time of a Markov chain is the minimum number of rounds needed to arrive at a distribution on Markov chain states that is ϵ-close to the stationary distribution.

Definition 7 ([23]): Let P be the transition matrix of a Markov chain with stationary distribution π, and let ρ be a stochastic vector. Then, the mixing time for ρ is defined as the number of rounds required for the Markov chain to approach within distance ϵ of the stationary distribution:

$$t_{mix}(\rho, \epsilon) := \text{argmin}_t\{\Delta(\pi, \rho \cdot P^t) \leq \epsilon\}$$

Here, Δ denotes the total variance distance (a distance metric for distributions defined in *Definition 8*). We also define:

$$t_{mix}(\epsilon) := \max_{\rho}\{t_{mix}(\rho, \epsilon)\}$$

When our scheme is instantiated with $R = t_{mix}(\epsilon)$ rounds, then it is an ϵ-secure sum-preserving encryption:

Lemma 2: Let ϵ be a negligible function of the security parameter. Then our scheme (Figure 4) using $R = t_{mix}(\epsilon)$ rounds satisfies NR-security.

Proof: Consider an adversary $\mathcal{A}$ attacking the NR-security of the scheme, *i.e.*, the adversary has access to an oracle $\mathsf{Enc}_K(\cdot, \cdot)$ and is nonce-respecting. This corresponds to the left-hand expression in *Definition 2*.

Now consider a hybrid interaction where all calls to the underlying PRF (in our scheme and also in the $n = 2$ scheme that is used as a component in our scheme) are replaced with uniformly random choices. This hybrid is indistinguishable to the adversary since all calls to the PRF are on distinct values (by the nonce-respecting property). In this hybrid, the output of encryption corresponds to a vector sampled according to the distribution $\rho \cdot P^R$ for some initial state ρ, where P is the Markov chain transition matrix.

By the assumption that $R = t_{mix}(\epsilon)$ and ϵ is negligible, we have samples that are indistinguishable from uniform samples in the Markov state space (*i.e.*, its stationary distribution). But uniform samples from the state space corresponds to the adversary $\mathcal{A}$ having oracle access to a random Φ-preserving mapping, as in the right-hand expression in *Definition 2*.

Overall, the adversary cannot distinguish between oracle access to Enc_K and a random Φ-preserving function. Hence, the scheme is NR-secure. ■

D. Theoretical Bound on Mixing Time

The definition of reversibility is included in Appendix (*Definition 11*). The P matrix corresponding to our scheme is not necessarily reversible. If the transition matrix P is non-reversible and has the stationary distribution $\pi = \{\pi_1, \pi_2, ..., \pi_{|\Omega|}\}$, the bound on mixing time features the second largest eigenvalue of a transition matrix M, which is intimately connected to the original matrix P. Consider the transition matrix $\overline{P}$ of the time-reversed chain to be $\overline{P} = D^{-1}P^TD$. In this equation, $D = diag\{\pi_1, \pi_2, ..., \pi_{|\Omega|}\}$. In this case, the matrix $M = P\overline{P}$ is reversible with respect to its stationary distribution, which is the same as the stationary distribution of P. On the other hand, the eigenvalues of M are real positive numbers smaller than or equal to 1 [7].

Lemma 3 ([7]): Let λ_* be the second-largest eigenvalue of $M = P\overline{P}$, where P is an irreducible, aperiodic transition matrix on the finite state space Ω. Then for any probability

distribution v on Ω, the following lemma holds:

$$|\nu^T P^t - \pi^T|^2 \leq {\lambda_*}^t \chi^2(\nu;\pi)$$

In this formula, t is the number of rounds and χ^2-contrast of ν with respect to π, is defined as follows:

$$\chi^2(\nu;\pi) = \sum_{x\in\Omega} \frac{(\nu(x)-\pi(x))^2}{\pi(x)}$$

Lemma 4: Let our non-reversible Markov chain have transition matrix P with $|\Omega|$ states, λ_* being the second-largest eigenvalue of the corresponding M. In this case, we can calculate an upper bound on the mixing time as:

$$t_{mix}(\epsilon) = \left\lceil \frac{2(\log\epsilon - \log(|\Omega|-1))}{\log\lambda_*} \right\rceil$$

Proof: This lemma has been proven in the Appendix. ■

E. Leveraging the Bound in Practice

Lemma 4 can potentially be leveraged to bound the mixing time of Markov chains associated with our construction in order to obtain a bound on the number of substitution and permutation iterations needed. There are three parameters in the bound as follows: ϵ, $|\Omega|$, and λ_*. ϵ can be fixed at 2^{-80} for 80-bit security. $|\Omega|$ and λ_* however, depend on the specific instance of the problem (*i.e.,* block size or the number of pixels, and the sum of the pixels in the block). Let a vector $\vec{v}$ with n elements and sum $s = \sum_i v$ be the starting configuration, where individual elements of the vector are bounded ($\mathbb{Z}_d$). If for this specific instance of the problem we calculate the P matrix, we can then calculate $|\Omega|$ and λ_* accordingly and compute the bound on mixing time. However, in real-world cases of interest (*e.g.,* 16×16 block of pixels), the size of the Markov chain is huge (*e.g.,* e^{1412} states in the Markov chain for 16×16 block of pixels with a sum of $\frac{16\times16\times255}{2}$) which makes calculation of the P matrix and consequently λ_* infeasible. Even with a powerful computer with dual sockets, 18-core Intel Xeon (R) processors (72 total VCPUs in total) and $256G$ RAM, memory limitations during the computation of eigenvalues prevented us from computing bounds for bigger instances of the problem.

We can explicitly calculate $|\Omega|$ for each instance of the problem (fixed $\vec{v}$, $\mathbb{Z}_d$, and s) using the following equation, when $n = |\vec{v}|$.

$$\sum_{k=0}^{n}(-1)^k\binom{n}{k}\binom{s-kd+n-1}{n-1}$$

Each term of the summation counts the number of *integer* vectors summing to s where at least k positions exceed $d-1$. Using a standard inclusion-exclusion technique, the overall sum counts the number of integer vectors summing to s where *no positions* exceed $d-1$ (*i.e.*, the number of vectors in $(\mathbb{Z}_d)^n$ with sum s).

It is easy to verify that for fixed n and $\mathbb{Z}_d$, the maximum of $|\Omega|$ appears when $s = \frac{nd}{2}$. Using this value in *Lemma 4* is of interest because we can reason about the worst case running time of our algorithm based on that. Figure 5 shows the relationship between the bound on mixing time and λ_* for different block sizes when the range is fixed to [0, 255] and the sum is set such that $|\Omega|$ is maximized. When λ_* is 0, all the rows of the P matrix are equal and the mixing time is 1, but when λ_* converges to 1, mixing time converges to ∞.

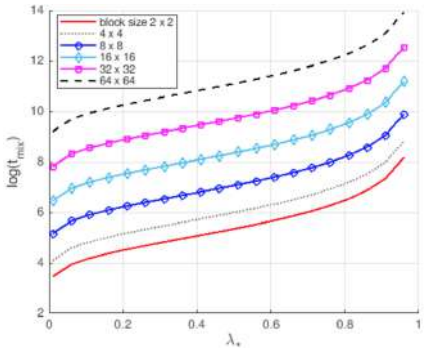

Fig. 5: The relationship between the bound on mixing time in log scale and λ_* for different block sizes. Pixel values range in [0, 255] and the sum is set such that $|\Omega|$ is maximized.

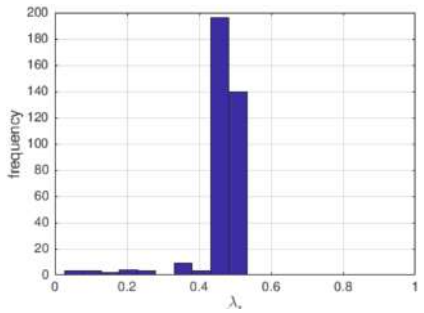

Fig. 6: Distribution of λ_* values for 363 small instances of our problem, where calculation of P matrix and λ_* are feasible.

The distribution of λ_* values for 363 small instances of our problem, where P and λ_* are calculable, is shown in Figure 6. We have explored d values belonging to the range [3,26], s values belonging to the range [2,148], and n values that are either 4 or 6. While this is a small sample, it is worth noting that we have not encountered cases where λ_* values are close enough to 1 to result in large mixing time bounds. In fact, we have not encountered a case where λ_* is higher than 0.54. It is also worth noting that in most cases the value of λ_* is close to 0.5. If λ_* remains more or less close to 0.5 in general cases, the bound on mixing time will be close to 2^{10} (about 1024 rounds) for a 32×32 block. Even if it were to reach 0.8 the bound on the number of rounds would be close to 2048. We explore the performance of our scheme with number of rounds in this range in the next section (Section VI). Further exploration of this distribution in general cases is an interesting problem and remains as future work.

VI. Prototype Implementation

A. Implementation

Web browsers are one of the most common interfaces to all the major cloud photo storage providers. As a proof-of-concept to demonstrate compatibility with existing services we implemented a Chrome browser plug-in which can be downloaded from the Chrome Web Store[7]. The goal of the plug-in is to help users work with TPE images and to provide the following functions:

[7]https://chrome.google.com/webstore/detail/auth/kkilcmjflngkjkhieceaahoppokkgeae?hl=en

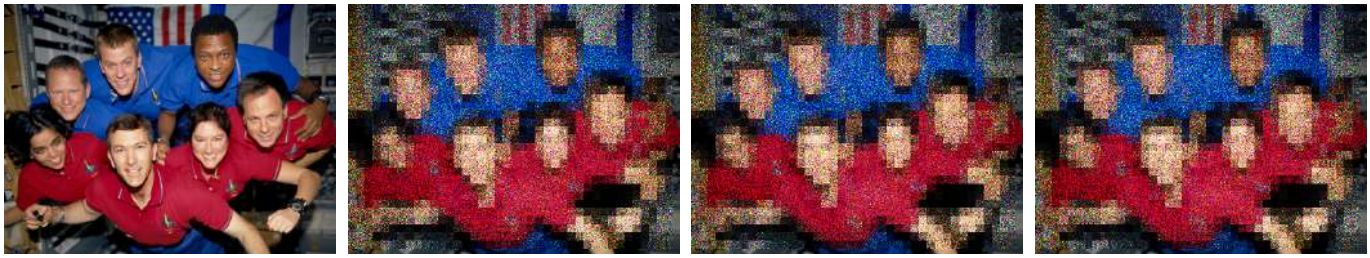

Fig. 7: TPE-encrypted images with constant block size (10×10) and varying number of iterations (Left to right: original, 100 iterations, 500 iterations, and 1000 iterations).

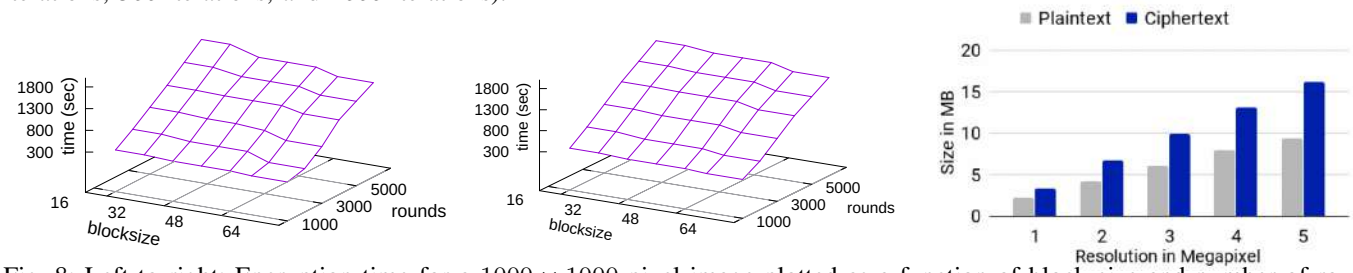

Fig. 8: Left to right: Encryption time for a 1000×1000 pixel image plotted as a function of block size and number of rounds. Decryption time for a 1000×1000 pixel image plotted as a function of block size and number of rounds. Image sizes for plaintext and ciphertext images.

Authentication: This is a one-time process. On the first use, the plug-in obtains an OAuth token for the cloud service provider. We used Google Photos in our case. The scopes for OAuth include *photoslibrary.appendonly*, which lets the plug-in add photos to the user's library, and *photoslibrary.readonly.appcreateddata*, which lets the plug-in read only the photos that were uploaded by the plug-in itself and not any other photos in the library.

Encrypt and upload: Users can use the plug-in to upload new photos to the cloud. These photos are encrypted locally in the browser with our TPE scheme and then uploaded to the user's cloud library. Users can browse through these encrypted photos on any device.

Decrypt and download: While browsing the photos, if the user decides to access (view or download) the original photo, the plug-in can download and decrypt the photo. Currently, the block size and iteration parameters are set in the plug-in configuration page and can be adjusted as needed.

B. Note on performance

Our TPE scheme is computationally expensive when compared to encrypting an image using standard AES. This is reflected in the observed encryption (decryption) times seen in Figure 8 where a 1 megapixel (MP) image takes about 280 seconds to encrypt (and similar time to decrypt) when using a block size of 50 and running for 1000 rounds. The encryption and decryption times increase more or less linearly with the number of rounds. Figure 7 shows a ciphertext TPE-encrypted with different number of rounds. It is easy to verify that while increasing the number of rounds changes minute details of the ciphertext, it does not impact the overall visual look of it.

Since each block can be encrypted independently, processing them in parallel should significantly reduce the encryption and decryption times experienced by end users. However, our current browser implementation is not multi-threaded as web JavaScript engines do not yet support multi-threading. A new web worker standard which will bring multi-threading to the web JavaScript engine is being developed by W3C. By using web workers' *ImageData* support we should see significant performance gains.

While, in theory, TPE does not increase the size of the ciphertext, in practice we did notice an increase in the size of ciphertext images of up to 73% when compared to the original image (see Figure 8). This is because formats like PNG use lossless compression schemes on raw images and cipertexts are harder to compress.

VII. Usability Evaluation

In this section, we evaluate end users' ability to work with TPE-encrypted images. Specifically, when users upload their photo album to the cloud, they expect to be able to browse through their images, distinguish between them, and identify and pick the desired ones when needed. We conducted a user study to compare user experience to do the aforementioned tasks in two different situations: with low-resolution thumbnails of the images as preserved by TPE versus the original images. Put another way, considering a dataset of images with which the users are already familiar, our goal in this user study is to pick a privacy-preserving thumbnail for each image and compare user experience with these thumbnails and original images. In order to do so, we compare two aspects of user experience: how accurately can users perform the given tasks (correctness scores) and how long it takes them (time).

With TPE, thumbnails can be preserved at a specified resolution, and the resolution of the thumbnail can function as a tuning parameter to trade-off between usability and privacy. In other words, the larger the thumbnail, the easier it is for users to browse, recognize, and distinguish images. However, larger thumbnails also mean they leak more information about the image, making it easier for unauthorized third parties to capture or infer private information from the image. For this study, we needed to pick a thumbnail size that users could

work with, while minimizing the information exposure. We discuss our approach to picking the thumbnail sizes later in Section VII-B.

Another important aspect for the validity of our study was to ensure that the users were familiar with the images, as would be the case in the real world scenario of browsing one's photo album. Our proposed approach to simulate users' familiarity is discussed next in Section VII-A

A. Photo Album Simulation

To simulate users' familiarity with images in their photo albums, we chose images from the popular TV series *Friends*. For study participants who were familiar with the characters and scenes from this TV series, images selected from *Friends* made a good substitute for images from users' own photo albums. Further, this approach allowed us to use the same image dataset for all participants and helped with the choice of thumbnail sizes as discussed in SectionVII-B.

Fig. 9: Five sample thumbnail images with different contexts (such as different setting and clothes) along with their corresponding original high-resolution versions.

Fig. 10: Five sample thumbnail images with same context (same clothes and same apartment setting) along with their corresponding original high-resolution versions.

Fig. 11: Six sample thumbnail portrait images from the six main characters of the series *Friends*, along with their corresponding original high-resolution versions.

B. Choice of Thumbnail Resolution

In order to pick a thumbnail size that reasonably minimizes information exposure while maximizing users' ability to browse, we leveraged a state-of-the-art image analysis platform, namely, Google's Cloud Vision API[8].

[8]https://cloud.google.com/vision/

We set a thumbnail resolution for each image as the largest thumbnail size where Google's Cloud Vision API was no longer able to recognize objects or faces in the image and could not provide meaningful information. For instance, the thumbnails shown in Figures 9, 10, and 11 are all at resolutions where Google's Cloud Vision API starts to fail. Thumbnail resolutions varied, due to variations in original image sizes and outputs from Google Vision API, from 42-64 pixels in width and 33-64 pixels in height. We followed this procedure to find the resolution threshold for 80 images and used the resulting thumbnails to evaluate users' ability to browse through and distinguish between images that were difficult for Google's Cloud Vision API to analyze correctly. While not using the participants' own images reduced ecological validity, this decision was made as uploading participants' images to Google's Cloud Vision API to check for thumbnail resolution threshold would impact their privacy unless the images are already publicly available.

C. Methodology

We recruited 200 participants through Amazon's Mechanical Turk (MTurk https://www.mturk.com/) to study the usability of TPE image thumbnails. We recruited another 100 participants to undertake the same study tasks but with high-resolution images as a control group. The survey was expected to take less than 25 minutes, although most participants finished the survey in a shorter amount of time. Participation was voluntary and participants were paid \$3.50 for their time. The study was approved by Oregon State University's institutional review board (IRB).

Participant Recruitment and Procedure: Our inclusion criteria were participants of age 18 years or older who had watched the TV series *Friends* and were familiar with the characters and the series in general. Participants were asked to give consent before proceeding to the actual user study. The study had several parts[9], which are discussed next.

Familiarity Test: In this part of the study, the participants were given nine high-resolution images from the series and were asked to name the characters they saw in each image. The purpose was to make sure that they are truly familiar with the main characters in the series and consequently a good candidate to participate in our study. These images were a good representative sample of the 80 images used in our study.

Matching Scenes with Descriptions (MSD): This test had four different parts as follows. As mentioned earlier, we conducted this study with both original images and thumbnails.

- *Matching 5 scenes with 5 descriptions* (MSD_{5D}): The users were asked to match 5 images with 5 descriptions, with the images being chosen from 5 different contexts. An example is shown in Figure 9. This test aimed to simulate a situation where a user is browsing through multiple images belonging to different albums and distinguishes between them.
- *Matching 5 scenes and 5 descriptions* (MSD_{5S}): This is a variation of the previous test. The difference is that the images all have the same context. This test simulates

[9]The complete survey questionnaire will be included in the full version of the paper.

browsing through images belonging to the same album and/or same setting. An example is shown in Figure 10.
- *Matching 10 scenes and 10 descriptions* (MSD_{10D}): This test is similar to MSD_{5D}, except that the number of images is increased to 10.
- *Matching 10 scenes and 10 descriptions* (MSD_{10S}): This test is similar to MSD_{5S}, except that the number of images is increased to 10.

Identifying a Scene Given a Description (ISD): This test also has four different parts similar to the MSD tasks.

- *Identifying a Scene Given a Description* (ISD_{5D}): The users were given 5 images and 1 description and were asked to pick the image that matched the description. The images were chosen from 5 different contexts. This test aimed to simulate a situation where a user is browsing through multiple images belonging to different albums and identifying a specific image of interest.
- *Identifying a Scene Given a Description* (ISD_{5S}): This is a variation of the previous test. Here, the 5 images were chosen from the same context, which simulates images belonging to the same album and/or setting.
- *Identifying a Scene Given a Description* (ISD_{10D}): This test is similar to (ISD_{5D}), except that the number of images is increased to 10.
- *Identifying a Scene Given a Description* (ISD_{10S}): This test is similar to ISD_{5S}, except that the number of images is increased to 10.

Portrait Character Recognition (PCR): In this experiment, participants were given 12 portrait images of the main characters, and were asked to name the characters. The goal was to evaluate a user's ability to distinguish between portrait images of familiar characters.

At the end of the study we compared how well users did when using original images and thumbnails both in terms of their correctness scores on each task/test, and how long they took in each case. Two One-Sided Tests (TOST) [15] is used for testing equivalence between the two distributions (see Section VII-D)

D. Study Findings

Familiarity: Users' familiarity was evident in the results of the familiarity test. The test scores were normalized to a scale of 0 to 1. Participants scored 0.89 on average (with a standard deviation of 0.2) indicating that the participants were indeed familiar with the characters of *Friends* television series and therefore good candidates to participate in the user study. We excluded one participant's data who said they were not familiar with *Friends*.

Matching Scenes with Descriptions (MSD): For scenarios MSD_{10S} and MSD_{10D} users were graded on a scale of 0 to 10 based on the number of their correct matches. Likewise, in sections MSD_{5S} and MSD_{5D} users are graded on a scale of 0 to 5 based on the number of their correct matches. For consistency, we have normalized all the scores to a scale of 0 to 1. Recognition times are captured in seconds. Table I shows average scores and times along with associated standard deviations for different MSD tests.

As can be expected, average correctness scores when using thumbnails are lower than when using original images (see Table I). The biggest drop 0.13 was found in the case of MSD_{10D}. However, the reductions in average scores are well within 0.5 standard deviation of average scores with original distributions. Further, using the TOST procedure we found that for MSD_{5S} and MSD_{5D}, if we allow for 0.1 difference in correctness scores (translated to 1 different answer or less out of the 5 answers), the distributions of scores with thumbnails and original images were found to be equivalent. Similarly, for MSD_{10S} and MSD_{10D}, if we allow for 0.15 and 0.2 difference in correctness scores respectively (translated to 2 different answers or less out of 10 for both tests), the distributions of scores with thumbnails and original images were found to be equivalent.

Similar trend to correctness was observed for the time taken for completing the tasks (see Table I). Time increased for all tasks with the biggest increase (about 36 seconds) observed in the case of MSD_{10S} (106.19 vs 89.54 seconds). However, all the increases were well within the standard deviations observed when working with original images. Using the TOST procedure, we found that the distributions of time taken with thumbnails and original images were found to be equivalent for MSD_{5S}, MSD_{5D}, MSD_{10S}, and MSD_{10D} if we allowed for 5, 15, 30 and 5 more seconds respectively.

Identifying scene for a given description (ISD): In all studied ISD scenarios, the participants were graded as either correct (1) or incorrect (0). Recognition times are captured in seconds. Table II shows average scores and times along with associated standard deviations for different ISD tests.

The average correctness scores when using thumbnails are lower than when using original images (see Table II) except for the case of ISD_{5D} where the score improved (0.87 vs 0.81). The biggest drop 0.15 was found in the case of ISD_{5S}. However, the reductions in average scores are well within or close to the standard deviation of average scores with original distributions. Further, using the TOST procedure we found that for ISD_{5S} and ISD_{5D}, if we allow for 0.05 and 0.15 difference in correctness scores (translated to 1 different answer or less out of the 5 answers), the distributions of scores with thumbnails and original images were found to be equivalent. Similarly, for ISD_{10S} and ISD_{10D}, if we allow for 0.2 and 0.1 difference in correctness scores respectively (translated to 2 different answers or less out of 10), the distributions of scores with thumbnails and original images were found to be equivalent.

The time taken to complete the tasks (see Table II) were close (less than 3 seconds difference) for all ISD tasks (well within standard devaitions) except for ISD_{10S} where it increased nearly 50% (18.09 vs 12.62 seconds) albeit still within the standard deviation. When using the TOST procedure, we found that the distributions of time taken with thumbnails and original images were found to be equivalent for all ISD tasks if we allowed for 5 more seconds.

Portrait character recognition (PCR): Participants answered 12 questions, each of which marked as either correct (1) or incorrect (0). The average of these scores is a number in the [0,1] range and used as participants' final score. Recognition times are captured in seconds. Table III shows the means

TABLE I: The means and standard deviations of correctness scores and times for different parts of MSD test performed on original and thumbnail images.

Test	correctness (μ, σ)	time (μ, σ)
MSD_{5S}-original	(0.93, 0.22)	(53.40, 38.66)
MSD_{5S}-thumbnail	(0.88, 0.26)	(78.22, 47.75)
MSD_{5D}-original	(0.94, 0.22)	(36.08, 45.75)
MSD_{5D}-thumbnail	(0.90, 0.27)	(38.43, 31.11)
MSD_{10S}-original	(0.91, 0.26)	(89.54, 57.15)
MSD_{10S}-thumbnail	(0.85, 0.30)	(106.19, 63.88)
MSD_{10D}-original	(0.89, 0.26)	(120.04, 64.13)
MSD_{10D}-thumbnail	(0.76, 0.31)	(155.82, 98.05)

TABLE II: Similar to Table I but for ISD.

Test	correctness (μ, σ)	time (μ, σ)
ISD_{5S}-original	(0.88, 0.10)	(19.31, 13.21)
ISD_{5S}-thumbnail	(0.74, 0.44)	(20.16, 16.65)
ISD_{5D}-original	(0.81, 0.38)	(18.87, 10.15)
ISD_{5D}-thumbnail	(0.87, 0.34)	(15.59, 12.75)
ISD_{10S}-original	(0.91, 0.28)	(12.62, 7.94)
ISD_{10S}-thumbnail	(0.80, 0.40)	(18.09, 30.81)
ISD_{10D}-original	(0.94, 0.24)	(12.83, 10.35)
ISD_{10D}-thumbnail	(0.91, 0.29)	(10.66, 7.44)

TABLE III: Correctness scores and times for PCR test performed on original and thumbnail images.

Test	correctness (μ, σ)	time (μ, σ)
PCR-original	(0.87, 0.26)	(55.82, 47.11)
PCR-thumbnail	(0.78, 0.24)	(59.87, 39.76)

and standard deviations of the scores and times.

As can be expected, average correctness score when using thumbnails is lower than when using original images (.78 vs. 0.87, see Table III). However, the reduction in average score is well within 0.5 standard deviation of average score with original images. Using the TOST procedure, we found that for PCR, if we allow for 0.15 difference in correctness score out of 1 (translated to 2 different answers or less out of the 12 answers), the distributions of scores with thumbnails and original images were found to be equivalent. The average times taken to complete the PCR task with thumbnails and original images were close (less than 5 seconds difference, see Table II). If we allowed for 15 more seconds (compared to 55.8 seconds) we had equivalence for PCR test.

The findings from our user study show that TPE has the potential to balance privacy and usability concerns. Overall, users did well in performing identification and matching tasks with thumbnail images (albeit they took a bit more time) when compared to original images - matching thumbnails with descriptions (average score 0.85 vs 0.92 out of 1), identifying a thumbnail from a given description (average score 0.83 vs 0.88 out of 1), and portrait character recognition (average score 0.78 vs 0.87 out of 1).

E. Study Limitations

While the results from this initial study are very promising and indicative of the usability of TPE images, caution should be exercised in generalizing the findings due to the following limitations. First, all participants did the recognition tasks in the same order, so the learning effect was not counterbalanced. Second, thumbnail sizes were picked to defeat one ML platform, namely Google's Cloud Vision API. It would be interesting to explore the lower limits for thumbnail resolutions where it becomes harder for users to work with them. Third, we also need to study how closely or well using images from a TV series mimics using images from a user's own photo album. We plan to undertake another study based on the results and lessons learned from this study to mitigate these limitations.

VIII. Related Work

Approaches like blurring, pixelation (mosaicing) or redaction have long been used for protecting privacy in images and text (*e.g.,* [49], [22], [43]). Simply blurring or pixelating an image is a destructive act that cannot be reversed. However, one can think of TPE as a reversible way to pixelate an image, since the ciphertext images reveal no more than a suitably pixelated version, in a cryptographic sense.

Studies of the privacy provided by pixelation/blurring techniques are therefore informative in determining suitable parameters for thumbnail block size in TPE. Indeed, pixelation with small block sizes is not very effective against both humans and automated recognition methods [49], [22], [27], [18], [26]. Specifically, it has been shown that it may be possible to deanonymize faces that are blurred or pixelated using ML techniques when block size (blur radius) is small and pictures of the person are otherwise available. Further, it has been shown that even if redaction techniques like replacing faces with white or black space are used, it may be possible to deanonymize when tagged pictures of the person are available using person recognition techniques [29].

Besides pixelation/blurring, reversible obfuscation techniques to protect image privacy have also been proposed (*e.g.*, [30], [39], [45], [48], [17], [46]). Many of these techniques (*e.g.*, [30], [39]) obfuscate the entire image (at least the public part). As a consequence, image owners are unable to distinguish between images without first decrypting them unless every image is tagged beforehand. Further, it has been shown that P3 [30] is vulnerable to face recognition with CNNs [26]. Cryptagram [39], on the other hand, bloats up the size of images significantly. Approaches that selectively obfuscate parts of an image, called region of interest (ROI) encryption, have also been proposed (*e.g.,* [47], [44], [45], [17], [46]). These approaches try to find the balance between utility, usability and privacy. However, the privacy and security guarantees typically lack cryptographic rigor, whereas ideal TPE provides an understandable cryptographic security guarantee.

Another approach for image privacy is the use of face de-identification (*e.g.,* [27], [13], [10], [19], [36]). Instead of obfuscating the image or parts of it, faces in the image are modified to thwart automated face recognition algorithms. These approaches extend the k-anonymity [37] privacy notion to propose the k-same privacy notion for facial images. At a high-level, to provide k-same protection for a face, it is replaced with a different face that is constructed as an average of k faces that are the most closest to the original face. These approaches have the benefit of providing a mathematically well-defined protection against face recognition algorithms. However, their downside is that they modify the original image and the transformation is not typically reversible. Further, while this approach protects against automated face recognition, it is not clear what level of usability this scheme has. A slightly different but related approach is where a face that

needs privacy protection in an image is replaced with a similar one from a database of 2D [6] or 3D faces [24].

Secure multi-party computation (SMC)-based approaches to image privacy (*e.g.,* [1], [34]) allow image processing without leaking unauthorized information to the parties. For instance, an entity Bob could process a databases of images belonging to Alice, without Alice learning his algorithm and without Bob learning about the images other than the output of the computation. However, besides being potentially expensive, SMC-based approaches require redesign of both the client and server in cloud-based multimedia applications.

Data or object removal approaches (*e.g.,* [21], [5], [31]) remove objects from images using *inpainting* techniques. They modify the original image in many cases irreversibly and are computationally intensive. A related approach is visual abstraction, where a face or an object is replaced with an abstract representations, such as a silhouette or a wire-line diagram (*e.g.,* [9], [38]). Again, these approaches are typically not reversible. To overcome this limitation, data-hiding approaches are proposed (*e.g.,* [8], [28]) where data and object removed from the image is encoded and stored in a hidden form (*e.g.,* through steganography) in the image itself or in an additional artifact (*e.g.,* in a cover video [42]). Such approaches either require service modification to deal with additional artifacts or require the original images or videos to have enough capacity to hide information and modify them irreversibly.

Zezschwitz *et al.* [40] have shown that techniques like pixelation at a high distortion value can be used to prevent image privacy loss from shoulder surfing by humans while still allowing image owners to browse through them on smart phones. As in their work we exploit the same human ability to recognize known images, objects, and faces even when they are distorted. As was noted in [40], this ability is influenced by what users know and have seen before [12] and this is known to be stronger if they themselves created (captured) the image [20]. In contrast to [40], TPE is applied to the original image, is keyed, and is completely reversible.

IX. Future Directions and Open Problems

Privacy and usability analysis: This work focused on qualitative evaluation of privacy and usability in the context of cloud-based photo storage services. Developing formal methods of quantitative analysis in other contexts remains as future work.

Computing explicit iteration bounds for ideal-TPE construction: We showed that our TPE scheme provides good security after sufficient number of rounds, and derived an upper bound for the necessary number of rounds. We were, however, not able to explicitly compute the round requirement (mixing time) for block sizes that are likely to be used in practice due to the extreme size of the underlying Markov transition matrices. We leave this as an open problem for future work.

Guidance for determining thumbnail size: To make TPE part of a truly usable system, it should be accompanied by guidance about how to select an appropriate thumbnail resolution. For instance, one goal of TPE is to allow designated users to identify or distinguish faces in their images (by inspecting the thumbnail) while preventing large-scale machine learning (ML) algorithms from doing the same. We used trial and error to determine thumbnail sizes that thwarted Google's Cloud Vision API. We would like a better understanding of how image size affects the effectiveness of modern facial-recognition approaches. To reach a desired failure rate of ML face-detection, how small (*e.g.,* how many pixels wide) must a face be? There already exists some relevant work on this question (*cf.* [26]), but the analysis is mostly on high-resolution images that give relatively high success rates for ML recognition.

Similarly, there is also prior work on reconstructing pixelated text [18], which can inform appropriate parameters when TPE is applied to text. Identifying the lower limits of thumbnail resolutions at which users are still able to work with different kinds of images (*e.g.,* faces and text), through a more comprehensive user study, is also an open problem.

Usability of TPE for other tasks: Our study explored the usability of TPE for tasks related to facial identification/recognition. It remains to be seen how usable TPE-encrypted images are in image classification tasks that do not involve faces.

Efficiency: Our Ideal-TPE construction, while novel, is expensive in terms of computation. While our construction uses a substitution step that considers 2 pixels at a time, one could design an efficient substitution function that considers, say, 4 pixels at a time instead. While this doesn't change the number of states in the associated Markov chain, it certainly increases the connectivity of the chain and intuitively should reduce the mixing time (through a reduction in the SLE value). We will explore this in an extended version of the paper. We also leave open the general problem of designing more efficient schemes (specifically, for sum-preserving encryption).

Full PRP security We have shown the feasibility of TPE that achieves the weaker NR definition of security. The PRP definition may be more desirable, as it makes the consequences of nonce-reuse less catastrophic. The challenge to achieving PRP security is that *every* plaintext thumbnail block must influence the encryption of *every other* thumbnail block. In contrast, for NR security it was enough to encrypt each thumbnail block independently.

If ciphertext expansion is allowed, then an approach inspired by the SIV construction [33] is likely to work: To encrypt M with nonce T, simply encrypt it under an NR-secure scheme with nonce $T^* := H(T\|M)$, where H is a collision-resistant hash function (or MAC). Intuitively, one cannot invoke the NR scheme with a repeated nonce without finding a collision under H. Unfortunately, this scheme requires knowledge of T^* to decrypt, hence leading to ciphertext expansion. We leave open the problem of achieving full PRP security without ciphertext expansion.

TPE for JPEG: In our construction, we assume "raw" pixels/images, yet the vast majority of captured images are encoded in JPEG format. Converting JPEG to raw image format for encryption and then encoding the ciphertext into JPEG will not work as JPEG encoding is lossy and can cause decryption to fail. Rather, one must operate directly in the native JPEG space, in which 8×8 image blocks are encoded in a frequency domain via discrete cosine transform. It turns out that the image thumbnail depends on only one of the coefficients in the frequency domain (the other coefficients determine the texture of the 8×8 block). One could adapt TPE to JPEG

by applying our TPE scheme to only these "DC" coefficients while encrypting all others with any efficient available format-preserving encryption scheme. Furthermore, in every 8×8 JPEG block there is only one DC coefficient. Hence, applying our scheme to 4×4 block of DC coefficients corresponds to a 32×32 block of the underlying image. This has the potential to significantly reduce TPE encryption costs, since fewer rounds may be required for the same thumbnail-block size. We will explore these ideas in our extended work.

X. Conclusion

Image privacy when storing images in the cloud is an important concern that will only grow. Traditional encryption schemes, including encrypted cloud storage, come with usability challenges as users cannot browse their images online without downloading and decrypting them. While image tagging and keyword-based image searches may alleviate this problem to an extent, it is not ideal for users to browse images with tags. The proposed thumbnail preserving encryption scheme is an attractive point in the design space where there is a trade-off of some privacy (we leak the thumbnail of the encrypted image) for usability. The trade-off between privacy and usability is tunable by controlling the size of the thumbnail leaked. Our evaluation shows that users are able to browse through TPE-encrypted images that leak very small thumbnails and that TPE can be integrated with existing services without any changes to those services. While the performance of our ideal-TPE construction needs to be improved, TPE is an interesting image encryption paradigm for balancing privacy and usability and opens many interesting directions for further research.

XI. Acknowledgements

We would like to thank Thinh Nguyen for introducing us to the concept of markov chain mixing time, and David Levin for the many insightful discussions. Also, we thank our study participants for generously offering their time and effort.

References

[1] Shai Avidan and Moshe Butman. *Blind Vision*, pages 1–13. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[2] Doug Beaver, Sanjeev Kumar, Harry C Li, Jason Sobel, Peter Vajgel, et al. Finding a needle in haystack: Facebook's photo storage. In *OSDI*, volume 10, pages 1–8, 2010.

[3] Mihir Bellare, Thomas Ristenpart, Phillip Rogaway, and Till Stegers. Format-preserving encryption. In Michael J. Jacobson Jr., Vincent Rijmen, and Reihaneh Safavi-Naini, editors, *Selected Areas in Cryptography, 16th Annual International Workshop, SAC 2009, Calgary, Alberta, Canada, August 13-14, 2009, Revised Selected Papers*, volume 5867 of *Lecture Notes in Computer Science*, pages 295–312. Springer, 2009.

[4] Tom Bergin. Yahoo email scanning prompts European ire. *Reuters*, October 2016.

[5] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 417–424, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[6] Dmitri Bitouk, Neeraj Kumar, Samreen Dhillon, Peter Belhumeur, and Shree K. Nayar. Face swapping: Automatically replacing faces in photographs. *ACM Trans. Graph.*, 27(3):39:1–39:8, August 2008.

[7] P. Bremaud. *Markov Chains: Gibbs Fields, Monte Carlo Simulation, and Queues*. Texts in Applied Mathematics. Springer New York, 2001.

[8] S.-C.S. Cheung, M.V. Venkatesh, J.K. Paruchuri, J. Zhao, and T. Nguyen. *Protecting and Managing Privacy Information in Video Surveillance Systems*, pages 11–33. Springer London, London, 2009.

[9] Kenta Chinomi, Naoko Nitta, Yoshimichi Ito, and Noboru Babaguchi. Prisurv: Privacy protected video surveillance system using adaptive visual abstraction. In *Proceedings of the 14th International Conference on Advances in Multimedia Modeling*, MMM'08, pages 144–154, Berlin, Heidelberg, 2008. Springer-Verlag.

[10] Benedikt Driessen and Markus Dürmuth. *Achieving Anonymity against Major Face Recognition Algorithms*, pages 18–33. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[11] Andy Greenberg. The police tool that pervs use to steal nude pics from Apple's iCloud. *Wired*, September 2014.

[12] Richard L. Gregory. Knowledge in perception and illusion. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 352(1358):1121–1127, 1997.

[13] Ralph Gross, Edoardo Airoldi, Bradley Malin, and Latanya Sweeney. Integrating utility into face de-identification. In *Proceedings of the 5th International Conference on Privacy Enhancing Technologies*, PET'05, pages 227–242, Berlin, Heidelberg, 2006. Springer-Verlag.

[14] Atsushi Harada, Takeo Isarida, Tadanori Mizuno, and Masakatsu Nishigaki. A user authentication system using schema of visual memory. In Auke Jan Ijspeert, Toshimitsu Masuzawa, and Shinji Kusumoto, editors, *Biologically Inspired Approaches to Advanced Information Technology*, pages 338–345, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[15] Walter W Hauck and Sharon Anderson. A new statistical procedure for testing equivalence in two-group comparative bioavailability trials. *Journal of Pharmacokinetics and Biopharmaceutics*, 12(1):83–91, 1984.

[16] Eiji Hayashi, Rachna Dhamija, Nicolas Christin, and Adrian Perrig. Use your illusion: Secure authentication usable anywhere. In *Proceedings of the 4th Symposium on Usable Privacy and Security*, SOUPS '08, pages 35–45, New York, NY, USA, 2008. ACM.

[17] J. He, B. Liu, D. Kong, X. Bao, N. Wang, H. Jin, and G. Kesidis. Puppies: Transformation-supported personalized privacy preserving partial image sharing. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 359–370, June 2016.

[18] Steven Hill, Zhimin Zhou, Lawrence Saul, and Hovav Shacham. On the (in)effectiveness of mosaicing and blurring as tools for document redaction. *Proc. Privacy Enhancing Technologies*, 2016(4):403–17, October 2016.

[19] A. Jourabloo, X. Yin, and X. Liu. Attribute preserved face de-identification. In *2015 International Conference on Biometrics (ICB)*, pages 278–285, May 2015.

[20] Hikari Kinjo and Joan Gay Snodgrass. Does the generation effect occur for pictures? *The American Journal of Psychology*, 113(1):95–121, 2000.

[21] A. C. Kokaram, R. D. Morris, W. J. Fitzgerald, and P. J. W. Rayner. Interpolation of missing data in image sequences. *IEEE Transactions on Image Processing*, 4(11):1509–1519, Nov 1995.

[22] Karen Lander, Vicki Bruce, and Harry Hill. Evaluating the effectiveness of pixelation and blurring on masking the identity of familiar faces. *Applied Cognitive Psychology*, 15(1):101–116, 2001.

[23] David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. *Markov chains and mixing times*. American Mathematical Society, 2006.

[24] Y. Lin, S. Wang, Q. Lin, and F. Tang. Face swapping under large pose variations: A 3d model based approach. In *2012 IEEE International Conference on Multimedia and Expo*, pages 333–338, July 2012.

[25] Byron Marohn, Charles V. Wright, Wu-chi Feng, Mike Rosulek, and Rakesh B. Bobba. Approximate thumbnail preserving encryption. In *Proceedings of the 2017 on Multimedia Privacy and Security*, MPS '17, pages 33–43, New York, NY, USA, 2017. ACM.

[26] R. McPherson, R. Shokri, and V. Shmatikov. Defeating Image Obfuscation with Deep Learning. *ArXiv e-prints*, September 2016.

[27] Elaine M. Newton, Latanya Sweeney, and Bradley Malin. Preserving privacy by de-identifying face images. *IEEE Trans. on Knowl. and Data Eng.*, 17(2):232–243, February 2005.

[28] Zhicheng Ni, Yun-Qing Shi, N. Ansari, and Wei Su. Reversible data hiding. *IEEE Transactions on Circuits and Systems for Video Technology*, 16(3):354–362, March 2006.

[29] Seong Joon Oh, Rodrigo Benenson, Mario Fritz, and Bernt Schiele. *Faceless Person Recognition: Privacy Implications in Social Media*, pages 19–35. Springer International Publishing, Cham, 2016.

[30] Moo-Ryong Ra, Ramesh Govindan, and Antonio Ortega. P3: Toward privacy-preserving photo sharing. In *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, pages 515–528, Lombard, IL, 2013. USENIX.

[31] A. RachelAbraham, A. Kethsy Prabhavathy, and J. Devi Shree. A Survey on Video Inpainting. *International Journal of Computer Applications*, 56(9):43–47, October 2012.

[32] Jordan Robertson. Dropbox confirms 2012 breach bigger than previously known. *Bloomberg*, August 2016.

[33] Phillip Rogaway and Thomas Shrimpton. A provable-security treatment of the key-wrap problem. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, volume 4004 of *Lecture Notes in Computer Science*, pages 373–390. Springer, 2006.

[34] Ahmad-Reza Sadeghi, Thomas Schneider, and Immo Wehrenberg. *Efficient Privacy-Preserving Face Recognition*, pages 229–244. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[35] Somini Sengupta and Kevin J. O'Brien. Facebook can id faces, but using them grows tricky. *The New York Times*, September 2012.

[36] Z. Sun, L. Meng, and A. Ariyaeeinia. Distinguishable de-identified faces. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 04, pages 1–6, May 2015.

[37] Latanya Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, October 2002.

[38] Suriyon Tansuriyavong and Shin-ichi Hanaki. Privacy protection by concealing persons in circumstantial video image. In *Proceedings of the 2001 Workshop on Perceptive User Interfaces*, PUI '01, pages 1–4, New York, NY, USA, 2001. ACM.

[39] Matt Tierney, Ian Spiro, Christoph Bregler, and Lakshminarayanan Subramanian. Cryptagram: Photo privacy for online social media. In *Proceedings of the First ACM Conference on Online Social Networks*, COSN '13, pages 75–88, New York, NY, USA, 2013. ACM.

[40] Emanuel von Zezschwitz, Sigrid Ebbinghaus, Heinrich Hussmann, and Alexander De Luca. You can't watch this!: Privacy-respectful photo browsing on smartphones. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 4320–4324, New York, NY, USA, 2016. ACM.

[41] Charles V. Wright, Wu-chi Feng, and Feng Liu. Thumbnail-preserving encryption for jpeg. In *Proceedings of the 3rd ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '15, pages 141–146, New York, NY, USA, 2015. ACM.

[42] Kenichi Yabuta, Hitoshi Kitazawa, and Toshihisa Tanaka. *A New Concept of Security Camera Monitoring with Privacy Protection by Masking Moving Objects*, pages 831–842. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[43] Jun Yu, Baopeng Zhang, Zhengzhong Kuang, Dan Lin, and Jianping Fan. iprivacy: Image privacy protection by identifying sensitive objects via deep multi-task learning. *IEEE Trans. Information Forensics and Security*, 12(5):1005–1016, 2017.

[44] L. Yuan, P. Korshunov, and T. Ebrahimi. Privacy-preserving photo sharing based on a secure jpeg. In *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 185–190, April 2015.

[45] L. Yuan, P. Korshunov, and T. Ebrahimi. Secure jpeg scrambling enabling privacy in photo sharing. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 04, pages 1–6, May 2015.

[46] Lin Yuan and Touradj Ebrahimi. Image privacy protection with secure JPEG transmorphing. *IET Signal Processing*, 11(9):1031–1038, 2017.

[47] Lin Yuan, David McNally, Alptekin Kupcu, and Touradj Ebrahimi. Privacy-preserving photo sharing based on a public key infrastructure, 2015.

[48] L. Zhang, T. Jung, C. Liu, X. Ding, X. Y. Li, and Y. Liu. Pop: Privacy-preserving outsourced photo sharing and searching for mobile devices. In *Distributed Computing Systems (ICDCS), 2015 IEEE 35th International Conference on*, pages 308–317, June 2015.

[49] Qiang Alex Zhao and John T. Stasko. The awareness-privacy tradeoff in video supported informal awareness: A study of image-filtering based techniques. Technical report, Goergia Tech, http://hdl.handle.net/1853/3452, 1998.

APPENDIX

A. Definitions, Lemmas and Proofs

Definition 8 ([23]): Total variation distance between two probability distributions μ and ν on Ω is as follows:

$$||\mu - \nu||_{TV} = 1/2 \sum_{x \in \Omega} \mu(x) - \nu(x)$$

Definition 9: A chain P is called irreducible, if for any two states $x, y \in \Omega$, there exists an integer t (possibly depending on x and y) such that $P^t(x, y) > 0$. This means that it is possible to get from any state to any other state using only transitions of positive probability [23].

Lemma 5: Markov chain model of our proposed TPE scheme is irreducible.

Proof: We develop a constructive method to show how to go from any state to any other state. Assume that the number of pixels in the block is k, and we want to go from configuration A = $a_1a_2...a_k$ to configuration B = $b_1b_2...b_k$ by using the substitution-permutation method. We show how to go from configuration A = $a_1a_2...a_k$ to configuration C = $b_1c_2...c_k$, where the first element of B is constructed. The construction of the next elements follows the same process. We consider the three following cases:

Case 1: $a_1 = b_1$: In this case, the first element is already constructed and we can go to the next stage to construct the second element.

Case 2: $a_1 < b_1$: In this case, we need to add the value of b_1-a_1 to the first element of A to construct the first element of B. This amount needs to be subtracted from other elements of A, namely $a_2...a_k$. We know that this amount is available to be subtracted, because if not, it means that $a_2 + ... + a_k < b_1 - a_1$ and this is a contradiction because it means that the sum of the elements in B, excluding b_1, is negative. In order to perform the subtraction of the value b_1-a_1, we set the position of a_1 to be fixed in the first position. Then we need to choose one element at a time from A, permute to bring it to the second position, where it can be substituted with a_1, and perform a substitution to add some value to the value of a_1. If the value added is not enough, then we perform another permutation and bring the next element to the second position and follow the same procedure until we reach b_1.

Case 3. $a_1 > b_1$: In this case, we need to subtract the value of a_1-b_1 from the first element of A to construct the first element of B. The construction is similar to the previous case, but instead of subtracting from the other elements of A, we should add to them. In this case, we need to make sure that adding to the elements in A will not cause any of them go out of range. Assuming that the range is represented by r, we would like to show that the following cannot be possible: $a_1 - b_1 > (r - a_2) + ... + (r - a_k)$. If this inequality is true, it means that the sum of the elements in B, excluding b_1, is more than $r \times (k - 1)$, which is a contradiction because the value of all the elements in B is bound by r. So the addition works and we can design the construction in the same way as

the previous case.

After we are done with constructing the first element of B, we can go ahead and do the same procedure for rest of the elements. In the next steps, the structure of the problem and the conditions remain exactly the same, making it feasible to continue applying the same method until we match the last two elements a_k and b_k. So, using a constructive proof, we showed that the chain is irreducible. ■

Definition 10: Let $T(x)$ be the set of times (rounds) when it is possible for the chain to return to starting position x. The period of state x is defined to be $gcd(T(x))$, *i.e.* the greatest common divisor of $T(x)$. The chain is aperiodic if all states have period 1. If a chain is not aperiodic, it is periodic [23].

Lemma 6: Our proposed Markov chain is aperiodic.

Proof: In our problem, we know P is aperiodic because we can go from each state x to itself after any desired number of rounds by simply staying still at each substitution and permutation round. So period of each state x is 1 and the Markov chain is aperiodic. ■

Lemma 7 ([23]): Let P be the transition matrix of an irreducible Markov chain. There exists a unique probability distribution π on Ω such that $\pi = \pi P$ and $\pi(x) > 0$, for all $x \in \Omega$.

Lemma 8 ([23]): Irreducible and aperiodic chains converge to their stationary distributions.

Lemma 9: For the Markov chain model of our TPE scheme, the uniform distribution on $\Phi(s)$ is the unique stationary distribution (hence the chain converges to this distribution).

Proof: Recall that the states of the Markov chain correspond to the elements of $\Phi(s)$ — i.e., vectors from $(\mathbb{Z}_d)^n$ whose sum (over the integers) is s. In this section we extend the notation and write $\Phi_n(s)$ to explicitly indicate the length (n) of the vectors.

One step of the Markov chain corresponds to one round of our encryption procedure, which itself consists of:

- Permuting the components of the state vector.
- Performing a random rank-encipher on adjacent pairs of components from the state vector.

It suffices to show that the uniform distribution over $\Phi_n(s)$ is invariant under both of these operations.

As the easiest case, consider sampling uniformly from $\Phi_n(s)$ and then permuting the components of that vector via *any fixed* permutation. This process induces a uniform distribution over $\Phi_n(s)$, since the definition of $\Phi_n(s)$ is symmetric with respect to the ordering within the vector. Similarly, applying an independently and uniformly chosen permutation to the vector induces a uniform distribution over $\Phi_n(s)$.

It is convenient to think of the rank-encipher step as *sequential* rank-encipher steps, one for each pair v_{2i-1}, v_{2i}. It suffices to show that the uniform distribution over $\Phi_n(s)$ is invariant to each one of these individual rank-encipher steps (by symmetry the one corresponding to v_1, v_2). When the entire vector $\vec{v}$ is sampled uniformly from $\Phi_n(s)$, the partial sum $t = v_1 + v_2$ follows some well-defined marginal distribution that we will denote as $\mathcal{T}_s$. Hence an alternative way to describe uniform sampling in $\Phi_n(s)$ is:

1) Sample $t \leftarrow \mathcal{T}_s$
2) Sample (v_1, v_2) uniformly from $\Phi_2(t)$.
3) Sample $(v_3, \ldots, v_n)$ uniformly from $\Phi_{n-2}(s-t)$.

The action of the rank-encipher step (after fixing its randomness) on (v_1, v_2) is that of a permutation over the set $\Phi_2(t)$, by correctness. But it is easy to see that sampling uniformly from $\Phi_2(t)$ (as in step 2), then applying a permutation to the result, still induces a uniform distribution. This is true no matter the distribution over $\Phi_2(t)$-permutations, as long as the distribution is independent of v_1, v_2 (as it is here).

This shows that the uniform distribution over $\Phi_n(s)$ is invariant to the rank-encipher step of our encryption algorithm, and hence the entire encryption algorithm. ■

Definition 11: A Markov chain P is reversible if the probability π on Ω satisfies the following condition for all $x, y \in \Omega$: $\pi(x)P(x,y) = \pi(y)P(y,x)$ [23].

Lemma 4: Let our non-reversible Markov chain have transition matrix P with $|\Omega|$ states, λ_* being the second-largest eigenvalue of the corresponding M. In this case, we can calculate an upper bound on the mixing time as:

$$t_{mix}(\epsilon) = \left\lceil \frac{2(\log \epsilon - \log(|\Omega| - 1))}{\log \lambda_*} \right\rceil$$

Proof: Our goal is finding $t_{mix}(\epsilon)$. This means that we want to find the smallest t for which the following equation is true for any initial distribution ν:

$$|\nu^T P^t - \pi^T| \leq \epsilon$$

We can square both sides and rewrite as:

$$|\nu^T P^t - \pi^T|^2 \leq \epsilon^2$$

Based on *Lemma 3* we know that the following equation holds, where t is the number of rounds:

$$|\nu^T P^t - \pi^T|^2 \leq {\lambda_*}^t \sum_{x \in \Omega} \frac{(\nu(x) - \pi(x))^2}{\pi(x)}$$

It therefore suffices to find a t such that:

$${\lambda_*}^t \sum_{x \in \Omega} \frac{(\nu(x) - \pi(x))^2}{\pi(x)} \leq \epsilon^2$$

Since our stationary distribution is uniform, we can replace $\pi(x)$ with $\frac{1}{|\Omega|}$. We can also replace $\nu(x)$ with 1 for all x, since doing so maximizes the sum and results in a $t_{mix}(\epsilon)$ that is bigger but still valid. Consequently, we can reform our equation as follows:

$${\lambda_*}^t \sum_{x \in \Omega} \frac{(1 - \frac{1}{|\Omega|})^2}{\frac{1}{|\Omega|}} \leq \epsilon^2$$

Based on this equation, we can retrieve the following equation:

$$t \geq \log_{\lambda_*}\left(\frac{\epsilon^2}{(|\Omega|-1)^2}\right) = \frac{2(\log \epsilon - \log(|\Omega|-1))}{\log \lambda_*} \tag{1}$$

Consequently, $t_{mix}(\epsilon)$ is calculated as follows:

$$t_{mix}(\epsilon) = \left\lceil \frac{2(\log \epsilon - \log(|\Omega| - 1))}{\log \lambda_*} \right\rceil$$

■

B. User Study Questionnaire

Note: Images shown in this sections, may have undergone minor resizing to improve the paper's readability.

1) Pre-Questionnaire:

Question 1: What is your age?

- 18 - 20
- 21 - 25
- 26 - 30
- 31 - 35
- 36 - 40
- 41 - 45
- 46 - 50
- 51 - 55
- 56 - 60
- 61 - 65
- 66 - 70
- 71 - 75
- 76 - 80
- 81 - 85
- 86 - 90
- 91 - 95
- 96 - 100
- 100+

Question 2: What is your gender?

- Male
- Female
- Other
- Prefer not to say

Question 3: What is your ethnicity?

- American Indian or Alaska Native
- Asian
- Black or African American
- Hispanic or Latino
- Native Hawaiian or Other Pacific Islander
- White
- Other
- Prefer not to say

Question 4: How many seasons of Friends have you watched?

- Have not watched
- Less than one (did not watch a full season)
- 1 - 3
- 4 - 7
- 8 - 10
- I have watched the whole series more than once

Question 5: How familiar do you think you are with the characters and the show in general?

- Very familiar
- Somewhat familiar
- Not familiar

Question 6: When was the last time that you watched the series or a part of it?

- Less than three months ago
- Three to six months ago
- Six months to a year ago
- A year to two years ago
- More than two years ago

2) Familiarity Test:

Prompt Presented to User: For each of the following images (See Figure 12), name all the characters you see in the image. You can select from the following list. Feel free to zoom in or out for any image. This could be done by right-clicking the image and selecting Open in New Tab to change the image size (Chrome or Firefox).

- Rachel
- Ross
- Phoebe
- Joey
- Monica
- Chandler

Follow-up Question: How much do you agree with the following statement? "Naming the character(s) in each image was difficult." *Presented to user after each of the nine questions in the familiarity test.*

- Strongly agree
- Agree
- Somewhat agree
- Neither agree nor disagree
- Somewhat disagree
- Disagree
- Strongly disagree

3) Matching Scenes with Descriptions (MSD):

Prompt Presented to User: For the following four sets of images, match each scene to the descriptions. Feel free to zoom in or out for any image. This could be done by right-clicking the image and selecting Open in New Tab to change the image size (Chrome or Firefox).

$\mathbf{MSD_{5S}}$: (See Figure 13)

- Phoebe and Chandler are about to kiss
- Monica and Chandler are next to each other, looking at Joey
- Phoebe and Chandler are talking
- Ross is pointing to himself
- Monica and Chandler are hugging, Phoebe is standing

$\mathbf{MSD_{5D}}$: (See Figure 14)

- Rachel, Monica, and Phoebe are in dresses, standing and talking
- Joey and Chandler are sitting and talking
- Rachel is standing and talking
- Rachel and Phoebe are sitting and talking
- Monica, Chandler, and Joey are on the beach

Fig. 12: Familiarity Test Images

Fig. 13: MSD_{5S} - Thumbnail (Top) & Control (Bottom)

Fig. 14: MSD_{5D} - Thumbnail (Top) & Control (Bottom)

$\mathbf{MSD_{10S}}$: (See Figure 15)

Fig. 15: MSD_{10S} - Thumbnail (Top) & Control (Bottom)

- Phoebe is standing
- Phoebe is talking to Joey at a restaurant
- Chandler and Monica are sitting
- Chandler and Joey are standing
- Chandler is standing and Joey is sitting
- Phoebe is sitting
- Rachel is sitting
- Joey is standing
- Joey is sitting
- Ross is standing

$\mathbf{MSD_{10D}}$: (See Figure 16)

- Chandler and Monica are at the beach
- Chandler and Monica are on a couch
- Joey and Monica are standing next to each other
- Chandler and Monica are in wedding clothes
- Chandler, Joey, and Phoebe are standing
- Ross is scaring Phoebe and Rachel
- Chandler and Joey are looking down
- Ross is taking a picture with Chandler
- Joey is talking to Rachel
- Rachel and Monica are standing

Fig. 16: MSD_{10D} - Thumbnail (Top) & Control (Bottom)

Follow-up Question: How much do you agree with the following statement? "Matching each description to an image was difficult." *Presented to user after each of the four questions in the MSD test.*

- Strongly agree
- Agree
- Somewhat agree
- Neither agree nor disagree
- Somewhat disagree
- Disagree
- Strongly disagree

4) Identifying a Scene Given a Description (ISD):

Prompt Presented to User: Pick an image that matches the description. Feel free to zoom in or out for any image. This could be done by right-clicking the image and selecting Open in New Tab to change the image size (Chrome or Firefox).

$\mathbf{ISD_{5S}}$: Monica is introducing her dollhouse (See Figure 17)

Fig. 17: ISD_{5S} - Thumbnail (Top) & Control (Bottom)

$\mathbf{ISD_{5D}}$: Monica and Rachel are talking to each other (See Figure 18)

Fig. 18: ISD_{5D} - Thumbnail (Top) & Control (Bottom)

ISD$_{10S}$: Ross is explaining something to Rachel (See Figure 19)

Fig. 19: ISD_{10S} - Thumbnail (Top) & Control (Bottom)

ISD$_{10D}$: Chandler and Monica are at the beach, looking at Joey (See Figure 20)

Fig. 20: ISD_{10D} - Thumbnail (Top) & Control (Bottom)

Follow-up Question: How much do you agree with the following statement? "Identifying a scene for a given distribution was difficult." *Presented to user after each of the four questions in the ISD test.*

- Strongly agree
- Agree
- Somewhat agree
- Neither agree nor disagree
- Somewhat disagree
- Disagree
- Strongly disagree

5) Portrait Character Recognition:

Prompt Presented to User: For each of the following images (See Figure 21), select the character you see in the image. You can select from the following list. Feel free to zoom in or out for any image. This could be done by right-clicking the image and selecting Open in New Tab to change the image size (Chrome or Firefox).

- Rachel
- Ross
- Phoebe
- Joey
- Monica
- Chandler

Follow-up Question: How much do you agree with the following statement? "Naming the character in each image was difficult." *Presented to user after each of the 12 questions in the portrait character recognition test.*

Fig. 21: Portrait Character Recognition Images

- Strongly agree
- Agree
- Somewhat agree
- Neither agree nor disagree
- Somewhat disagree
- Disagree
- Strongly disagree

6) Post-Questionnaire:

Question 1: What Operating System did you use?

Question 2: Did you open and view images in a new tab and/or window?

- Yes
- No
- Other (Please explain)

Question 3: Did you find that a specific way of viewing the image files (e.g. zooming in/out or changing the viewing mode) was more helpful to answer the questions? If yes, please describe the specific way you viewed.

Question 4: Would you feel comfortable posting an image online if its thumbnail was of a similar quality as the following examples (See Figure 22), and if this thumbnail was the only thing that anyone can see without your explicit permission to see the full resolution image? *This question was only asked in thumbnail user study.*

(a) Example 1

(b) Example 2

Fig. 22: Post-Questionnaire 4

Question 5: Is there anything else about your experience with the survey/images that you would like to share with us?