ABSTRACT

Facing a large number of personal photos and limited resource of mobile devices, cloud plays an important role in photo storing, sharing and searching. Meanwhile, some recent reputation damage and stalk events caused by photo leakage increase people's concern about photo privacy.

Images are becoming one of the key enablers of user connectivity in social media applications. Many of them are directly exploring image content to suggest new friends with similar interests. To handle the explosive volumes of images, one common trend is to leverage the public cloud as their robust service backend.

Despite the convenience, exposing content-rich images to the cloud inevitably raises acute privacy concerns. In this paper, we propose a privacy-preserving architecture for image-centric social discovery services, designed to function over encrypted images.

The proposed system is a new technique by using multi-secret sharing as the underlying encryption, which indeed induces a blow-up issue of the key size. For preserving the efficiency of the key size, we apply a compression by using lightweight cryptographic algorithms. This scheme based on the proposed techniques, and show effectiveness, efficiency, and security by experiments and analysis.

CHAPTER NO	TABLE OF CONTENTS	PAGE NO

LIST OF FIGURES

LIST OF ABBRIVATIONS

RDH	Reversible Data Hiding
RDHEI	Reversible Data Hiding Encrypted Image
LSB	Least Significant Bit
SIK	Share Independent Secret Keys
SNK	Share No Secret Key
SOK	Shared One Key
OAMSS	Operating Addition homomorphism in multi-Secret Sharing
MCGs	Mirroring CipherText Groups
PRF	Probabilistic Random Forest
LSH	Locality Sensitive Hashing
RAD	Rapid Application Development
RAM	Random Access Memory
HTML	Hyper Text Markup Language
IDE	Integrated Development Environment
CFS	Correlation Based Feature

CHAPTER 1

INTRODUCTION

Reversible data hiding (RDH) is a notion that allows to embed the additional and secret message into cover media, such as military or medical images, and to perform a reversible procedure that extracts the hidden secret message and perfectly reconstructs the original cover content. Numerous reversible data hiding methods have been introduced over the last two decades. Two seminal ideas of RDH are difference expansion (proposed by Tian [1]) and histogram shifting (proposed by Ni et al. [2]). In the difference expansion method [1], the differences between two adjacent pixels are doubled to release a new least significant bit (LSB) plane for carrying the secret message. In the histogram shifting method [2], the zero and peak points are used to embed the secret message by slightly modifying the pixel values. Many RDH studies have elaborated these two concepts to improve payload and image quality [3, 4, 5, 6, 7, 8, 9]. Recently, a new direction of RDH known as RDH over an encrypted image (RDHEI) has been introduced. This novel RDHEI notion was firstly introduced by Zhang in 2011 [10], and captures the following real-life scenario regarding owner privacy known as image privacy [10]. An inferior assistant or a channel administrator is in the middle of a workflow and is authorized to insert some additional data such as the origin information, image notations or authentication data, within the encrypted image, where the original image content is unknown to this party. Indeed, medical images are encrypted for preserving the patient privacy, and a database administrator only embeds a few data into the corresponding encrypted images. For the consistency of a medical image, it must guarantee that the original content can be perfectly reconstructed after decryption-then-extraction of the secret message by the receiver. That is, RDHEI not only ensures the accuracy of the reconstructed cover-image and extracted secret message which are two basic tasks of RDH, but also preserves the privacy of the cover-image. More precisely, the work of Zhang [10] formalizes the model to describe the aforementioned scenario. The image provider P intends to preserve the privacy of the cover-image, but still desires a data hider H to embed a secret message.

Therefore, H embeds the message into the encrypted image which is generated by P from the cover-image. Finally, the receiver R can recover the original cover-image and then extract the secret message correctly. The procedure run by R is known as decryption-then-extraction. However, the receiver also can be divided into two steps (decryption and extraction). We specify these two steps to two kinds of receivers, Rdec and Rext, and Rdec performs decryption, and Rext takes Rdec's decrypted image to extract the secret message. The amount of digital images has increased rapidly on the Internet. Image security becomes increasingly important for many applications, e.g., confidential transmission, video surveillance, military and medical applications. For example, the necessity of fast and secure diagnosis is vital in the medical world.

In this paper, we develop an effective and reliable framework for RDH in the encrypted domain. In fact, the proposed method belongs to the third category. Its main contribution is the combination of the modular addition and two-dimensional (2D) histogram modification. Its advantages are mainly manifested in four aspects. First of all, room for data hiding does not need to be vacated before encryption, which is more reasonable compared with the methods in [25–28]. Secondly, completely separable and completely reversible can be achieved, which is more reliable than the methods in [18–21]. Thirdly, the modular arithmetic addition operation, which has additive homomorphism, is utilized for image encryption. It does not cause data expansion, unlike the public-key cryptosystems in [29, 31–33]. Finally, since data embedding in encrypted domain is accomplished by using pairwise coefficient modification, embedded capacity has been greatly improved compared with the methods in [30, 34]. The rest of the paper is organized as follows. In Section 2, we describe the proposed scheme, which includes image encryption, data embedding in encrypted image, data extraction, and original image recovery. Experimental results and analysis are presented in Section 3. Finally, in Section 4, conclusions and future work are drawn.

1.1 RELATED WORK

A comprehensive survey on RDH is presented by Shi et al. [11] to deeply analyze and highlight the advances of RDH for the recent progress. It studies aspects of RDH, including RDH into image spatial domain [1, 2], RDH into image compressed domain (e.g., JPEG) [12, 13, 14], RDH suitable for image semi-fragile authentication [15,16,17], etc. In particular, it also investigates RDHEI, and categorizes the existing RDHEI schemes into two classes: vacating room before encryption and vacating room after encryption by embedding strategies. For key setting, Shi et al. [11] also mentioned the other notion, so-called RDHEI based on public key encryption. However, inspired by the factor of key setting, the present studies identify the following two notions of RDHEI.

Numerous insightful works have proposed this type of RDHEI schemes. _ Share no secret key (SNK). In contrast to SIK, R does not need to share any secret key. This can be easily achieved through public key encryption where R has a public/secret key pair, and P (H, resp.) can use the public key to do image encryption (data embedding, resp.). The first solution, proposed by Chen et al. [9], is to use Paillier homomorphic encryption [10] to encrypt each pixel and rely on specific techniques to complete data embedding. With the use of the homomorphic encryption, the follow-up works of Zhang at al. [11], Li and Li [12], and Shiu et al. [13] respectively implement some reversible data hiding techniques under the public key encryption associated with the homomorphic property.Tsai et al. [3] present a predictor, may be denoted local difference (LD) predictor, that computes difference between pixels intensities in a local area of the image and most central one in the area to bring out prediction-errors.

They embed secret data via histogram modification of the prediction errors. Sachnev et al. [4] further present cross-dot predictor that divides image into two “cross” and “dot” sets. Dot set may be predicted using cross one and vice versa. Cross-dot predictor is also represented as chess-board (CB) predictor in [5]. In general, the content owner expects to send only an encrypted image to the manager without extra information. In addition to VRAE and RRBE, another type of method has recently been proposed by using homomorphic encryption. With the additive homomorphic property of Paillier cryptosystem, Chen et al. [15] frstly

proposed a homomorphic encryption based RDH approach. Shiu et al. [17] improved Chen et al.'s method by adopting the concept of diference expansion into homomorphic encryption.

Moreover, RDH in the homomorphic encrypted domain has also been investigated in [18]. However, the used public-key cryptosystems lead to data expansion afer image encryption. In [16], the additive homomorphic property of modulo operation is utilized to realize the RDH in the encrypted domain.

1.1.1 Share independent secret keys (SIK)

R shares independent keys, keyP and keyH, with P and H respectively. Notably, these keys (keyP; keyH) are secret and used to run image encryption and embedding algorithms. Numerous insightful works [10, 18, 19, 20, 21, 22] have proposed this type of RDHEIschemes.

1.1.2 Share no secret key (SNK).

In contrast to SIK, R does not need to share any secret key. This can be easily achieved through public key encryption where R has a public/secret key pair, and P (H, resp.) can use the public key to do image encryption (data embedding, resp). The first solution, proposed by Chen et al.[23], is to use Paillierhomomorphic encryption [24] to encrypt each pixel and rely on specific techniques to complete data embedding.

With the use of the homomorphic encryption, the follow-up works of Zhang at al. [25], Li and Li [26], and Shiu et al. [27] respectively implement some reversible data hiding techniques under the public key encryption associated with the homomorphic property. To summarize the flexibility of key setting, it is clear that only the designated party who has the secret key can be P or H in SIK. However, the advantage of SNK is that anyone can be P or H, since the keys of encryption or embedding are exactly the public key. In addition, as known, those homomorphic encryption-based SNK-type RDHEI schemes are practically inefficient since the underlying encryption schemes usually rely on complicated algebra structures and spend high computational cost. It suffices to give the following question, and we will aim for addressing it in the remainder of this paper. In fact, Wu et al. [28] had proposed a shared one key (SOK) scheme based on

secret sharing. However, their method spends much space cost, since it encrypts a pixel into n shares, where n is the security parameter of secret sharing, and the total cost of an encrypted pixel will blow up to 8nbits.

1.2 MAIN CONTRIBUTIONS

Let us briefly summarize our results. Our starting point is to formalize the new notion of key setting and care about. The key setting offers the framework among P, H, and R. For example, if a RDHEI scheme is under public key encryption, anyone can be P and H. If under a symmetric encryption between P and R (H and R, resp.), only specific party who holds the shared secret key can be P (H, resp.). The key use is referred to as party flexibility. In the following, we formalize the key setting for more details. The efficiency to achieve better efficiency, we must avoid using public key encryption. However, if we do not use any public key encryption scheme, it is impossible to protect image privacy (the original purposes of RDHEI) without the shared key between P and R. Thus, for preserving privacy, the ideal class is that receiver shares “only one” secret key with the image provider (SOK, for short). In particular, there is no shared key between H and R, which precisely implies that the embedding procedure does not take any shared key as input. The proposed SOK schemes are inspired from some existing SNK schemes (i.e., [25, 27]). We found that these SNK schemes work with Paillier encryption (or other addition homomorphic encryption) to preserve image privacy, and the property of homomorphic evaluation is used to embed the message. For achieving our above-mentioned requirements, we replace the parts of Paillier encryption with secret sharing that also enjoys homomorphic evaluation in some ways2. We show an abstraction of those SNK schemes, and then under the abstraction, introduce our method. The high level idea of our method is composed of the following two steps.

Image recovery If the encryption key is available on the recipient side, the encryption key can be used to decrypt the original image. The sender sends the encryption key to the recipient. This key is used as the start value of the feedback shift register. This initial value is used to generate a pseudorandom number that is added to the pixel value of the image and performs the modular operation with 256. The generated value will be the new pixel value. The same procedure is

performed for all pixels in the cover image. Therefore, the image is decrypted. Pseudo-random numbers are generated by subtracting a random number from a starting value. The result is placed at the end of the starting value by moving the starting value to the left. D. Data extraction Data extraction is the reverse process of data integration. Initially, the key is included in the encrypted image where the data is displayed. Group LSB bits into 5 bits. Then check 5bits based on this control symbol with control symbols 1B, 1C, 1D and 1E. Finally, the original data is acquired. Encryption

Secret sharing acts as a symmetric encryption to encrypt the cover-image, so our method use one shared key between P and R. However, ours does not construct shares for each pixel like Wu et al.'s method. For preserving the total size, we pack t pixels and t random factors together to generate only t shares, and put the shares back as encrypted pixels and set random factors as the key. It suffices to avoid the size blow-up, and also keeps correctness of decryption by using t random factors and t shares. The technique of our method is inspired by the multi-secret sharing, but we slightly modify it for security and framework of SOK.

1.2.1 Data Embedding

Data embedding is a new steganographic method for combining digital information sets. This paper describes the data embedding method and gives examples of its application using software written in the C-programming language. Sandford and Handel produced a computer program that implements data embedding in an application for digital imagery. To provide security for the embedded data, one can remove the key from the combined data and manage it separately. The image key can be encrypted and stored in the combined data or transmitted separately as a ciphertext much smaller in size than the embedded data. The data embedding method applies to host data compressed with transform, or `lossy' compression algorithms, as for example ones based on discrete cosine transform and wavelet functions. Analysis of the host noise generates a key required for embedding and extracting the auxiliary data from the combined data. The key is stored easily in the combined data. Images without the key cannot be processed to extract the embedded information.