Denoising in the Dark: Privacy-Preserving Deep Neural Network-Based Image Denoising

Yifeng Zheng, Huayi Duan, Xiaoting Tang, Cong Wang, *Senior Member, IEEE*, and Jiantao Zhou

Abstract—Large volumes of images are being exponentially generated today, which poses high demands on the services of storage, processing, and management. To handle the explosive image growth, a natural choice nowadays is cloud computing. However, coming with the cloud-based image services is acute data privacy concerns, which has to be well addressed. In this paper, we present a secure cloud-based image service framework, which allows privacy-preserving and effective image denoising on the cloud side to produce high-quality image content, a key for assuring the quality of various image-centric applications. We resort to state-of-the-art image denoising techniques based on deep neural networks (DNNs), and show how to uniquely bridge cryptographic techniques (like lightweight secret sharing and garbled circuits) and image denoising in depth to support privacy-preserving DNN based image denoising services on the cloud. By design, the image content and the DNN model are all kept private along the whole cloud-based service flow. Our extensive empirical evaluation shows that our security design is able to achieve denoising quality comparable to that in plaintext, with high cost efficiency on the local side and practically affordable cost on the cloud side.

Index Terms—Image denoising, cloud computing, privacy preservation, deep neural networks

1 Introduction

Large volumes of images are being exponentially generated today from various kinds of imaging devices (smartphones, medical imaging equipment, digital cameras, etc.). Such tremendous growth is further greatly accelerated along with the fast development and wide deployment of diverse Internet of Things (IoT) applications. For example, in IoT multimedia healthcare applications [1], images are continuously generated from IoT-enabled medical imaging devices [2], which range from being large and stationary [3], to being small, portable, and handheld [4], [5]. Such tremendous growth has led to high demands on the services of image storage, processing, and management, which are now naturally addressed via the adoption of cloud computing [6], [7], [8], [9], [10]. One fundamental and long-standing image processing task is image denoising, which refers to the procedure that takes as input a noisy image and produces a denoised image where the noise is reduced [11]. In practical situations, image noises are often hard to avoid in the first place [12], since they can come from different internal (i.e., imaging devices) and external (i.e., environment) sources [13], [14]. With the ability to remove noises and deliver high-quality image content, image denoising typically serves as a necessary stepping stone for many other applications like image segmentation, image registration, image super-resolution, and more [14], [15].

In the literature, state-of-the-art image denoising techniques rely on deep neural networks (DNNs), where a well-trained DNN model is used to map a noisy image to a denoised one [16]. In order to achieve high denoising quality, the DNN model used for denoising is typically large, containing a number of hidden layers with thousands of neurons on each layer [16]. This poses high computation workload, and thus is especially suitable to be handled at the cloud for its abundant yet economical computing resources. Despite the appealing benefits, leveraging the cloud for DNN based image denoising also entails critical privacy concerns. First, the images generated from various application contexts could be proprietary and/or could be privacy-sensitive, like medical images and faces images. Second, the DNN model could be proprietary information of the model owner, and may also leak information of the underlying possibly sensitive training images [17]. Therefore, it is of paramount importance to enforce the protection of images and of the DNN model, against the cloud throughout the whole service flow.

In light of the above observations, in this paper, we present the first system framework enabling privacy-preserving DNN based image denoising services on the cloud. Our framework enables DNN based image denoising to be conducted on the cloud side, while keeping the (noisy/denoised) images and DNN model private. That is, along the whole service flow, the cloud only sees ciphertexts of the (noisy/denoised) images and DNN model. While in theory fully homomorphic encryption [18], [19] could be used to achieve such security features and functionality, it is way far

• Y. Zheng and C. Wang are with the Department of Computer Science, City University of Hong Kong, Hong Kong, China, and also with the City University of Hong Kong Shenzhen Research Institute, Shenzhen 518057, China. E-mail: yifeng.zheng@my.cityu.edu.hk, congwang@cityu.edu.hk.
• H. Duan is with the Department of Computer Science, City University of Hong Kong, Hong Kong, China. E-mail: hduan2-c@my.cityu.edu.hk.
• X. Tang is with the Department of Computer Science, Brown University, Providence, RI 02912 USA. E-mail: tang_xiaoting@brown.edu.
• J. Zhou is with the Department of Computer and Information Science, Faculty of Science and Technology, and also with the State Key Laboratory of Internet of Things for Smart City, University of Macau, Macau 999078, China. E-mail: jtzhou@umac.mo.

Manuscript received 8 Oct. 2018; revised 24 Jan. 2019; accepted 18 Mar. 2019. Date of publication 25 Mar. 2019; date of current version 13 May 2021. (Corresponding author: Cong Wang.)
Digital Object Identifier no. 10.1109/TDSC.2019.2907081

1545-5971 © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: POLO BIBLIOTECARIO DI INGEGNERIA. Downloaded on June 02,2021 at 15:51:59 UTC from IEEE Xplore. Restrictions apply.

inefficient for practical use [20], [21], and would induce prohibitive runtime and communication cost on both local side and cloud side if applied to our case. The challenge we aim to tackle is how to achieve privacy-preserving DNN based image denoising with effective denoising quality, high local cost efficiency, and practically affordable performance on the cloud. To this end, we bridge together cryptography and image denoising in depth, so as to securely, effectively, and efficiently embrace the operations required by DNN based image denoising.

First, to simultaneously achieve privacy protection and high local cost efficiency, we resort to the lightweight cryptographic technique, i.e., additive secret sharing, to encrypt (noisy) images. Then, encrypted DNN based image denoising is conducted over the secret shares of images on the cloud side. In our framework, we consider that the cloud entity is split into two cloud servers which are hosted by independent cloud service providers. Such a two-server model is commonly used in various security application contexts in recent years (e.g., [22], [23], [24], [25], to just list a few), and we consider our adoption also to be in this trend. Specifically, in our framework each cloud server respectively holds secret shares of noisy images and the DNN model. And a secure protocol highly customized for DNN based image denoising is run between the two cloud servers to produce the ciphertexts of denoised images.

To accomplish this secure protocol, a strawman solution is to directly use garbled circuits, which is a well-known secure two-party computation protocol enabling secure evaluation of arbitrary functions. However, directly applying garbled circuits to fully complete our target complicated DNN based image denoising would lead to highly burdensome performance overhead. This is because in garbled circuits even a simple function can lead to a circuit with an excessive number of gates [26]. In fact, motivated by this, many recent security works choose to work under the blueprint of using a hybrid approach: bridging together garbled circuits and other secure computation techniques to build custom protocols and solve application-specific tasks (e.g., [26], [27], [28], to just list a few).

Following the same blueprint, in this paper we show how to uniquely bridge together additive secret sharing and garbled circuits to deliver a highly customized design for privacy-preserving DNN based image denoising. Indeed, garbled circuits are used very selectively in our design (Section 4). Through in-depth examination on the procedure of DNN based image denoising, we first carefully decompose the procedure and identify the underlying atomic operations. Then we consider how to properly leverage additive secret sharing and garbled circuits to embrace these operations in a secure, efficient, and customized manner. Specifically, in our design, as an instantiation, we use the multi-layer perceptron (MLP) model proposed in [16] as the concrete DNN model, which is one of the most representative and state-of-the-art DNN models for image denoising. We identify that the atomic operations turn out to be addition, multiplication, and evaluation of a non-linear activation function, i.e., the hyperbolic tangent function. For secure addition and secure multiplication, we rely on specialized use of additive secret sharing, and work over secret-shares under a vectorized setting customized for DNN based image denoising, which inherently requires operations among matrices/vectors [16], [29]. Regarding secure evaluation of the activation function, we note that it requires division and exponentiation in the ciphertext domain, which is hard to be practically supported [30].

To ease this tension, our idea is to seek a secure-computation-friendly and high-accuracy approximation for the hyperbolic tangent function in DNN based image denoising. In particular, we resort to a piecewise non-linear and low-degree polynomial approximation with high accuracy and very small quantitative approximation errors [31], where exponentiation and division are avoided. Based on this approximation, we then develop a highly customized mechanism to support secure and efficient evaluation of the approximated function for image denoising, via a tailored combination of additive secret sharing and garbled circuits. At a very high level, we rely on decomposing approximated function into atomic operations, and mostly compute on secret-shared values. Then, garbled circuits is only used when the comparison operation is required. In this way, we achieve the best of both worlds: additive secret sharing for efficient arithmetic operations and garbled circuits for efficient Boolean operation.

To facilitate the practical usage of our design, we also further explore and address several practical considerations, including number representation, secure shared-value rescaling, local pre-processing and post-processing, and communication cost reduction. We implement a proof-of-concept prototype and conduct an extensive evaluation over a real-world image dataset. Our experiments results show that the denoising quality achieved by our security design is comparable to that in plaintext. Besides, it is validated that our security design has highly efficient local cost and practically affordable cloud-side cost. Altogether, we make the following contributions:

- We present the first system framework enabling privacy-preserving DNN based image denoising services on the cloud. Our framework allows DNN based image denoising to be conducted on the cloud side, while keeping the image content and the DNN model private along the whole service flow.
- We present a security design highly customized for DNN based image denoising. Our design uniquely bridges lightweight additive secret sharing and garbled circuits, with secure and efficient mechanisms for customizing the atomic operations underlying DNN based image denoising.
- We implement a proof-of-concept prototype and conduct a comprehensive evaluation. The evaluation results show that our security design achieves denoising quality comparable to that in plaintext, with high local-side cost efficiency and practically affordable cloud-side cost.

1.1 Related Work

Privacy-Preserving Image Denoising. In the literature, there exist some works on privacy-preserving image denoising in cloud computing [32], [33]. We identify that the common philosophy underlying the image denoising approach of these works is to produce denoised image patches via appropriately aggregating a set of reference patches, which can be selected either from external databases [32] or from inside the noisy image itself [33]. In [32], Zheng et al. study how to achieve

Authorized licensed use limited to: POLO BIBLIOTECARIO DI INGEGNERIA. Downloaded on June 02,2021 at 15:51:59 UTC from IEEE Xplore. Restrictions apply.

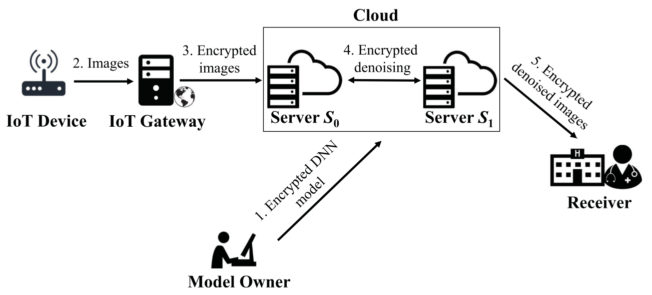


Fig. 1. The illustration of our system architecture.

privacy-preserving image denoising from encrypted external databases hosted by the cloud. In particular, they consider a setting where a service provider outsources an encrypted database of image patches to the cloud, which is then required to offer secure query based image denoising service to authorized users. In [33], Hu et al. propose a double-cipher scheme for non-local means based encrypted image denoising. In their scheme, two ciphertexts are generated for each noisy image. One ciphertext is generated via (expensive) homomorphic encryption for supporting encrypted mean filter, and the other is obtained by a distance-preserving transform function, so as to support non-local search. Despite being valuable data points in the design space of privacy-preserving image denoising in cloud computing, we emphasize that these designs consider denoising techniques different from state-of-the-art DNN based image denoising targeted by us. Very recently, Zheng et al. [34] present a first study on DNN based image denoising with security features, which heavily uses homomorphic encryption. Their work substantially differs from ours in two aspects. First, they target a different problem setting where a user stays online to interact with the cloud server throughout procedure of denoising of his images. In contrast, in our system architecture all parties (the IoT gateway, receiver, and model owner) are not required to stay online to participate in the encrypted denoising procedure with the cloud. Second, their security design has very cumbersome performance overhead: a single DNN computation in their security design can take up to tens of hours.

Secure Deep Neural Network Inference. Our work is also related to existing works on secure DNN inference, which generally aims to leverage a *trained* DNN model for data inference in a secure and privacy-preserving manner [30], [35], [36], [37], [38], [39], [40], [41]. The concrete application setting of most of these works [30], [35], [36], [37], [38], [39], [40], however, is different from ours. Specifically, these works operate in a two-party setting where a client holding private data wishes to leverage a trained DNN model held by a server to get inference results. The overall security goal is to ensure that the client only learns the inference results (without learning the model) while the server learns nothing about the client's data. So, in these works, the trained model is held in the clear by the server. In contrast, in our application setting, the model is concealed to the cloud servers, and they only see along the whole workflow ciphertexts of the neural network model, the network input, and the network output.

Moreover, we emphasize that those prior works have different problem focuses and their security designs are highly customized to serve their purposes. For example, the work [35] focuses on the square activation function and builds a highly customized design for the problem of image classification; the works [30], [37], [40] focus on considering customized designs for different non-determined and non-closed-form approximation of activation functions which requires manual and empirical parameter tuning; the work [41] targets image classification and relies on a determined approximation for the sigmoid function to propose a customized security design for image classification.

The rest of this paper is organized as follows. Section 2 presents our problem statement. Section 3 describes some preliminaries. Section 4 gives our detailed design of privacy-preserving DNN based image denoising. Section 5 presents the experiment results. Section 6 concludes the whole paper.

2 Problem Statement

2.1 System Model

Our work targets privacy-preserving DNN based image denoising on the cloud. To facilitate the presentation, we will use denoising for images generated from IoT multimedia healthcare [1], [2] applications as a concrete example to demonstrate our system design. Fig. 1 shows the system architecture that enables IoT multimedia healthcare applications with cloud-based privacy-preserving DNN based image denoising. There is an IoT gateway [42], [43], [44] that connects the IoT devices to the cloud. The IoT gateway can be operated by a hospital or a medical center, which continuously collects images from the IoT devices in various healthcare contexts, and then outsources the storage, management, and analytics of the images to the cloud. For privacy protection, the images will be first encrypted by the IoT gateway before being outsourced to the cloud.

In practical situations, due to various intrinsic (i.e., sensor) and extrinsic (i.e., environment) conditions, the images collected from the IoT devices are easy to be corrupted with noise [12]. Hence, in order to ensure the quality of IoT healthcare applications, it is highly demanded to let the cloud perform image denoising so as to deliver high-quality image content for accurate medical diagnosis and/or for any further image analytic tasks. For high-quality image denoising, in our system architecture, we will resort to the approach based on DNNs, which has been shown to achieve the state-of-the-art denoising quality in recent years [16], [29]. Specifically, in our system architecture an encrypted trained DNN model will be stored in the cloud in advance by the model owner (e.g., a medical lab that has expertise in DNN model training). After the encrypted images are sent to the cloud by the IoT gateway, image denoising is performed in the encrypted domain based on the pre-stored encrypted DNN model. The denoised images in encrypted form can later be delivered on demand to the receiver (e.g., a physician in the hospital).

The cloud entity in our system architecture is split into two cloud servers $\mathcal{S}_0$ and $\mathcal{S}_1$, which can be hosted by independent cloud service providers in practice. We note that such a two-server model has been widely adopted in the literature under various security application contexts (e.g., [22], [24], [41], to just list a few) and our adoption also follows this popular trend. Particularly, the two cloud servers will jointly perform the task of high-quality DNN based image denoising in the encrypted domain. In our protocol design, for fast image encryption on the IoT gateway side, we will resort to the

Authorized licensed use limited to: POLO BIBLIOTECARIO DI INGEGNERIA. Downloaded on June 02,2021 at 15:51:59 UTC from IEEE Xplore. Restrictions apply.

TABLE 1
Key Notations Used in This Paper

Notations	Description
$\mathbf{p}$	Noisy patch
$\mathbf{q}$	Denoised patch
$\mathbf{G}_j$	Weight matrix of the jth layer of the DNN model
$\mathbf{b}_j$	Bias vector of the jth layer of the DNN model
$\mathbf{I}$	Noisy image
$\mathbf{D}$	Denoised image
$\mathbf{R}$	Random matrix used in secret sharing
$[\mathbf{C}]_i$	Image ciphertext held by cloud server $\mathcal{S}_i$
σ	Standard deviation of noise
l	Ring size parameter
s	Scaling factor parameter

lightweight cryptographic technique called additive secret sharing. In particular, for each (noisy) image, the IoT gateway generates two secret shares, and sends each secret share to each cloud server. Besides, each cloud server also holds a secret share of the DNN model in advance. Based on the secret shares of the collected images and DNN model, the two cloud servers jointly run an encrypted DNN based image denoising protocol to produce the encrypted denoised images.

2.2 Threat Model

In this paper, we consider that the security threats primarily come from the adoption of cloud services. That is, the two cloud servers are considered as adversaries. Following most of existing security works under the two-server model (e.g., [41], [45], to just list a few), we assume a semi-honest and non-colluding security model in this paper. In particular, the two cloud servers will honestly follow our protocol for encrypted DNN based image denoising, yet they are interested in inferring private information about the image content and DNN model and will do so independently.

We note that such a security assumption also makes sense in our target cloud-assisted image services, as in practice cloud service providers are well-established and business-driven parties and they are not willing to risk their highly valuable reputation by behaving maliciously and colluding with each other [22], [24]. Under such a security assumption, our security goal is to ensure that the image content and DNN model are all kept private against the two cloud servers along the whole service flow. Note that the DNN model also demands protection in practice as it could be proprietary information of the model owner, and may also leak information about the training data which might be privacy-sensitive [30].

3 Preliminaries

3.1 DNN Based Image Denoising

Image denoising is a well-known and fundamental problem in image processing, which, roughly speaking, aims to map a noisy image to a noise-free image. Mathematically, a noisy image $\mathbf{I}$ is generally modeled as

$$\mathbf{I} = \mathbf{X} + \mathbf{E}, \tag{1}$$

where $\mathbf{X}$ is the original clean image, and $\mathbf{E}$ is the additive white Gaussian noise (AWGN) with standard deviation σ [16], [29], [46], [47], [48], [49]. Given a noisy image $\mathbf{I}$, image denoising aims to produce an estimate of the original image $\mathbf{X}$ as accurately as possible. Note that here σ can be estimated from a noisy image through various effective existing methods (e.g., [50], [51], [52]) and thus in the literature it is treated as a known prior and serves as an input to a denoising algorithm (e.g., [16], [46], [53]). So, we also consider σ as a known priori. We stress that the noise assumptions simply follow plaintext-domain works on image denoising and our focus is on crafting a privacy-preserving design for DNN based image denoising which is introduced below.

For high denoising quality, we resort to state-of-the-art DNN based image denoising techniques in this paper. Specifically, we will use the widely popular DNN model proposed in [16] as an instantiation, which is one of the most representative and state-of-the-art DNN models for image denoising [29], with relatively simple structure and outstanding denoising performance. In [16], it is proposed that a multi-layer perceptron can be used as a DNN model for high-quality denoising. Specifically, given an input vector $\mathbf{p}$, an MLP is essentially a nonlinear function that maps $\mathbf{p}$ to the output vector $\mathbf{q}$, via feeding the input to a pipeline of layers. For example, an MLP with $L-1$ hidden layers can be written as,

$$\mathbf{q} = \mathbf{G}_L \cdot tanh(...tanh(\mathbf{G}_1\mathbf{p} + \mathbf{b}_1)...) + \mathbf{b}_L), \tag{2}$$

where $\mathbf{G}_j$ denotes the weight matrix and $\mathbf{b}_j$ the bias vector, of the jth layer ($j \in [1, L]$). DNN based image denoising is performed patch-wise [16], [29], so here the input vector $\mathbf{p}$ actually represents a patch of the noisy image and the output vector $\mathbf{q}$ is the denoised patch. Therefore, for DNN based image denoising, a noisy image is first divided into overlapping patches, and each patch is then denoised separately. Afterwards, we can place the denoised patches at the locations of their noisy counterparts and produce the denoised image via aggregating the overlapping region [16].

3.2 Yao's Garbled Circuits

Yao's garbled circuits enables two parties to compute a function f on their respective data, in such a manner that the parties' inputs are kept private against each other and only the function output is revealed at the end [54], [55]. Specifically, one party acts as a generator who first generates a garbled version of the circuit computing $f(g_1, g_2)$, where g_1 is this party's input and g_2 is the other party's input. The generator sends the garbled circuit and the garbled input $\widehat{g_1}$ of g_1 to the other party, which is called the evaluator. The evaluator runs an oblivious transfer protocol [56] with the generator to obliviously obtain the garbled input $\widehat{g_2}$ of his private input g_2. With $\widehat{g_1}$ and $\widehat{g_2}$, the evaluator evaluates the garbled circuit to obtain the target result $f(g_1, g_2)$.

In Table 1, we provide a summary of the key notations used in this paper.

4 Privacy-Preserving DNN Based Image Denoising

In this section, we will elaborate on our design for enabling privacy-preserving DNN based image denoising. At a high level, the workflow in our system architecture is as follows. First, the DNN model is encrypted by the model owner (e.g., a medical lab) and the resultant ciphertexts are sent to the two cloud servers for later use in denoising. Subsequently,

Authorized licensed use limited to: POLO BIBLIOTECARIO DI INGEGNERIA. Downloaded on June 02,2021 at 15:51:59 UTC from IEEE Xplore. Restrictions apply.

the IoT gateway continuously collects images from the IoT devices, encrypts each image, and sends the ciphertexts to the two cloud servers. After receiving the ciphertexts of each image, the two cloud servers jointly run a secure protocol to perform DNN based image denoising in the encrypted domain, and produce the ciphertexts of the denoised image. Later, the two cloud servers can deliver the ciphertexts of denoised image on demand to the receiver, who then performs decryption to recover the denoised images. Along this workflow, our protocol design can be decomposed into three core components accordingly, i.e., (i) encryption of images and DNN model, (ii) encrypted DNN based image denoising, and (iii) decryption of encrypted denoised images. In what follows, we first present in detail each component. Then, we analyze the security guarantees of our design, and address some practical considerations.

4.1 Encryption of Images and DNN Model

For practical cost efficiency, we resort to the lightweight cryptographic technique—additive secret sharing, for the encryption of images and DNN model. We note that secret sharing based encryption can result in ciphertexts which are twice the size of plaintexts. However, this is still substantially less than the common homomorphic encryption approach for supporting encrypted computation, which can cause up to tens of times size expansion (say from 64-bit plaintext to 2048-bit ciphertext in the widely adopted Paillier cryptosystem) [57].

In particular, given an image $\mathbf{I}$, the IoT gateway first generates a random matrix $\mathbf{R}$, where each element of $\mathbf{R}$ is uniformly random in $\mathbb{Z}_{2^l}$ and l is the parameter deciding the ring size. Then, the IoT gateway produces the ciphertexts of the image as $[\mathbf{C}]_0 = (\mathbf{I} - \mathbf{R}) \bmod 2^l$ and $[\mathbf{C}]_1 = \mathbf{R} \bmod 2^l$, and sends $[\mathbf{C}]_0$ and $[\mathbf{C}]_1$ to the two cloud servers $\mathcal{S}_0$ and $\mathcal{S}_1$ respectively. We denote the secret share held by cloud server $\mathcal{S}_i$ as $[\mathbf{C}]_i$, where $i \in \{0, 1\}$. Similarly, for the DNN model, each cloud server $\mathcal{S}_i$ receives $\{[\mathbf{G}_j]_i\}_{j=1}^{L}$ and $\{[\mathbf{b}_j]_i\}_{j=1}^{L}$ in advance. Note that the encryption of the DNN model only needs to be conducted once, and afterwards the encrypted DNN model can be deployed at the cloud side for denoising.

4.2 Encrypted DNN Based Image Denoising

Given the secret shares of an image and the DNN model, we now describe how to perform encrypted DNN based image denoising over the secret shares on the cloud. The result from the encrypted image denoising procedure is that each cloud server obtains a secret share of the denoised image. In what follows, we will first describe how to securely support the underlying atomic operations, and then present the full construction. Note that unless otherwise stated, all arithmetic operations related with secret shares take place in $\mathbb{Z}_{2^l}$ and for ease of presentation we will omit the modulo operation in the description of our design.

4.2.1 Design Overview

We now first consider how to securely support in our design the atomic operations underlying DNN based image denoising over secret-shared values. According to Eq. (2) in Section 3.1, it is not difficult to see that the atomic operations underlying DNN based image denoising include addition, multiplication, and evaluation of the non-linear activation function $tanh(\cdot)$. First of all, we consider how to support secure addition and secure multiplication over secret-shared values, as a starting point.

Supporting Secure Addition and Multiplication. Given the secret shares of two values a and b, secure addition can be performed to enable each cloud server to obtain a secret share of the sum $a + b$, without disclosing a and b to the two cloud servers. For this operation, we can simply let each cloud server $\mathcal{S}_i$ $(i \in \{0, 1\})$ locally compute $[a + b]_i = [a]_i + [b]_i$. Regarding secure multiplication, the goal is to properly multiply the secret shares of a and b so that each cloud server $\mathcal{S}_i$ can obtain $[a \cdot b]_i$ without knowing a and b. We note that this can be achieved via leveraging the Beaver's technique [58], which is described as follows. Suppose that a multiplication triplet (x, y, z) is secret-shared among the two cloud servers, where x and y are random values and $x \cdot y = z$. Each cloud server $\mathcal{S}_i$ first locally computes $[u]_i = [a]_i - [x]_i$ and $[v]_i = [b]_i - [y]_i$. Then, each cloud server $\mathcal{S}_i$ broadcasts $[u]_i$ and $[v]_i$, and they reconstruct u and v. Now each cloud server $\mathcal{S}_i$ computes the share $[a \cdot b]_i = i \cdot u \cdot v + u \cdot [y]_i + v \cdot [x]_i + [z]_i$. We refer the readers to [58] for the proof of correctness and the extensions to more than two servers. Note that for the simple case of multiplication where one input value is a public constant c, we can simply let each cloud server $\mathcal{S}_i$ compute $[c \cdot b]_i = c \cdot [b]_i$.

Customizing Secure Addition and Multiplication. To achieve better efficiency in DNN based image denoising which inherently requires operations among matrices and/ or vectors, we generalize the above secure operations to work under the vectorized setting for matrices and/or vectors, inspired by some latest advancements [30], [41], [59]. In particular, given two shared matrices $[\mathbf{A}]$ and $[\mathbf{B}]$, we can compute addition as $[\mathbf{A} + \mathbf{B}]_i = [\mathbf{A}]_i + [\mathbf{B}]_i$. To compute secure multiplication, we now need the multiplication triplet in a vectorized form. Specifically, given the secret shares of three matrices $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$, where $\mathbf{X}$ has the same size as $\mathbf{A}$, $\mathbf{Y}$ has the same size as $\mathbf{B}$, and $\mathbf{Z} = \mathbf{X} \cdot \mathbf{Y}$, we can compute the secret share of $\mathbf{A} \cdot \mathbf{B}$ as follows. First, each cloud server $\mathcal{S}_i$ computes $[\mathbf{U}]_i = [\mathbf{A}]_i - [\mathbf{X}]_i$ and $[\mathbf{V}]_i = [\mathbf{B}]_i - [\mathbf{Y}]_i$. Then, they jointly reconstruct $\mathbf{U}$ and $\mathbf{V}$. Afterwards, each cloud server $\mathcal{S}_i$ locally computes $[\mathbf{A} \cdot \mathbf{B}]_i = i \cdot \mathbf{U}\mathbf{V} + \mathbf{U}[\mathbf{Y}]_i + [\mathbf{X}]_i\mathbf{V} + [\mathbf{Z}]_i$. Note that the secret shares of data-independent multiplication triplets can be made available to the two cloud servers via a trusted dealer or via an existing cryptographic protocol in an *offline* phase [41], [59] before actual secure multiplication takes place. So, throughout this paper, we assume that the multiplication triplets are available on the cloud side for use, and our main focus is on the online encrypted denoising procedure.

Supporting and Customizing Secure Evaluation of the Activation Function. Besides secure addition and multiplication, we also need to consider how to securely support the evaluation of the non-linear activation function $tanh(\cdot)$. We note that the evaluation of the $tanh(\cdot)$ function requires exponentiation and division, which are hard to be supported using secure computation techniques [30], [41]. Therefore, our insight is to seek a secure computation friendly approximation for the $tanh(\cdot)$ function, so that it can be efficiently supported by secure computation techniques. While it is known that in principle we can use high-degree polynomials for function approximation [41], [60], this may not be a good choice for

Authorized licensed use limited to: POLO BIBLIOTECARIO DI INGEGNERIA. Downloaded on June 02,2021 at 15:51:59 UTC from IEEE Xplore. Restrictions apply.

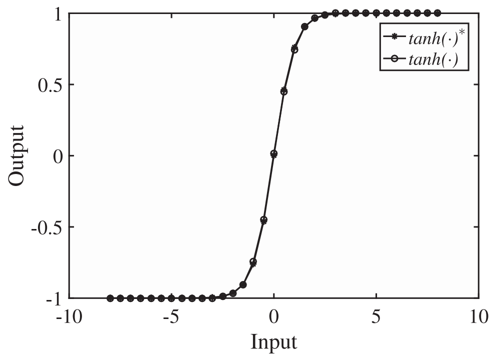


Fig. 2. The $tanh(\cdot)$ function and its approximation in this paper.

practical cost efficiency and low-degree polynomials are preferable for secure computation [30]. Hence, we resort to low-degree polynomial approximation and particularly consider the technique of piecewise polynomial approximation.

Instead of using linear approximation which requires empirical and heuristic tuning on the number of pieces and also typically a large number (say more than 12) of pieces [30] for high accuracy, we resort to non-linear low-degree polynomial approximation and particularly the promising approximation in [31] which has a closed-form solution and *quantitative* approximation performance. Fig. 2 illustrates the approximation $tanh(\cdot)^*$ adopted in this paper for the $tanh(\cdot)$ function. The details of the $tanh(\cdot)^*$ function are as follows:

$$\tanh(x)^* = \begin{cases} k \times (n_1|x|^2 + c_1|x| + d_1), 0 \leq |x| \leq a \\ k \times (n_2|x|^2 + c_2|x| + d_2), a < |x| \leq b \\ k, \text{otherwise.} \end{cases} \quad (3)$$

where $k = sign(x)$, and n_1, n_2, c_1, c_2, d_1, d_2, a and b are -0.2716, -0.0848, 1, 0.42654, 0.016, 0.4519, 1.52, and 2.57, respectively. This approximation has quantitative performance: the average error and maximum error of this approximation are 4.1×10^{-3} and 2.2×10^{-2} respectively [31]. Note that the parameters in this approximation are obtained via solving an error minimization problem, which is orthogonal to our security design. Our focus is to delicately build on this approximation and explore a new design point to support privacy-preserving DNN-based image denoising on the cloud. Additionally, we remark that in principle the closed-form approximation for the $tanh(\cdot)$ function is also applicable to the Sigmoid function, due to the inherent relationship between the $tanh(\cdot)$ function and the Sigmoid function, i.e., constant scaling and shifting.

Based on the above effective and efficient approximation, we then consider how to securely and efficiently evaluate the approximate activation function $tanh(\cdot)^*$. Note that by saying the approximation is effective, we mean that it has small approximation errors; and by saying it is efficient, we mean that the computation required by the approximation would be simple (only addition, comparison, and multiplication are needed) and lightweight to be supported in the encrypted domain, as opposed to the expensive and complicated exponentiation and division operations in the original function. As comparison operation is required in the approximation, we proposed to leverage garbled circuits which can securely and efficiently handle comparison operation [28], [41]. Given the secret shares of a value x, an intuitive mechanism of using garbled circuits for the evaluation of $tanh(x)^*$ is as follows. We can use garbled circuits to securely reconstruct the original value x from its secret shares, perform the function evaluation inside the garbled circuit, and secret-share the computation result among the two cloud servers again for subsequent processing. In more detail, we can let $\mathcal{S}_1$ provide $\mathcal{S}_0$ with a garbled circuit inside which their secret shares are combined to recover the original value, and then the $tanh(\cdot)^*$ function is evaluated against the recovered value. The evaluation of the garbled circuit on the $\mathcal{S}_0$ side will output a secret share $[tanh(x)^*]_0 = tanh(x)^* - r$, where $tanh(x)^*$ is the function evaluation result and r is a random value chosen by $\mathcal{S}_1$. The corresponding secret share for $\mathcal{S}_1$ is $[tanh(x)^*]_1 = r$.

Despite being effective, such an intuitive mechanism is not efficient as it requires multiplication inside the garbled circuit. Note that in garbled circuits even simple functions may lead to an excessive number of gates. For example, the multiplication of two l-bit values will require $O(l^2)$ gates [26], [28]. Therefore, for good efficiency, we choose to use garbled circuits very selectively. Specifically, for secure evaluation of the $tanh(\cdot)^*$ function in our security design, our main idea is to conduct all polynomial-related computation (where multiplication is required) under secret sharing outside the garbled circuit, and only do addition (for recovery of original values) and comparison (for selection of polynomial computation result) inside the garbled circuit. In this way, we obtain the best of both worlds: additive secret sharing for efficient arithmetic operation and garbled circuits for efficient Boolean operation.

4.2.2 Our Construction

We now present the detailed construction of privacy-preserving DNN based image denoising on the cloud side. Recall that as introduced in 3.1, DNN based image denoising is performed patch-wise, so in what follows we will show how to denoise a patch of a noisy image, given the secret shares of the patch and the secret shares of the DNN model. To ease our presentation, we will present our protocol for the case of one hidden layer for demonstration. That is, $\mathbf{q} = \mathbf{G_2} tanh(\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{b}_1)^* + \mathbf{b}_2$. The extension to more layers is quite natural, as the output of this layer serves as input of the next layer, followed by the same necessary operations (i.e., secure addition, secure multiplication, and secure evaluation of the approximated activation function). Note that all intermediate and final results are secret-shared between the two cloud servers. Suppose that the patch to be denoised is $\mathbf{p}$ and each cloud server $\mathcal{S}_i$ holds a secret share $[\mathbf{p}]_i$. The details of our construction are as follows.

First, the two cloud servers need to compute the secret shares of $\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{b}_1$. Suppose that we have the multiplication triplet $\mathbf{X}_1, \mathbf{y}_1, \mathbf{z}_1$, where $\mathbf{X}_1$ has the same dimension as $\mathbf{G}_1$, $\mathbf{y}_1$ has the same dimension as $\mathbf{p}$, and $\mathbf{z}_1 = \mathbf{X}_1 \cdot \mathbf{y}_1$. The computation of the secret shares of $\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{b}_1$ is given in Fig. 3. Subsequently, we need to obtain the secret shares of $tanh(\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{b}_1)^*$. As mentioned before, we handle the polynomial computation outside the garbled circuit so as to avoid within-circuit multiplication operations. So, inside the garbled circuit, we only need to do the operations of modular addition and comparison. From Eq. (3), we know that for the case $x > 0$, we need to compute two values $t_1 = n_1 x^2 + c_1 x + d_1$ and $t_2 = n_2 x^2 + c_2 x + d_2$; for the case

Authorized licensed use limited to: POLO BIBLIOTECARIO DI INGEGNERIA. Downloaded on June 02,2021 at 15:51:59 UTC from IEEE Xplore. Restrictions apply.

```
Input: $[\mathbf{G}_1]_i$, $[\mathbf{p}]_i$, $[\mathbf{b}_1]_i$, $[\mathbf{X}_1]_i$, $[\mathbf{y}_1]_i$, and $[\mathbf{z}_1]_i$.
Output: $[\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{b}_1]_i$
  1: $\mathcal{S}_i$ computes $[\mathbf{U}_1]_i = [\mathbf{G}_1]_i - [\mathbf{X}_1]_i$ and $[\mathbf{v}_1]_i = [\mathbf{p}]_i - [\mathbf{y}_1]_i$.
  2: $\mathcal{S}_0$ and $\mathcal{S}_1$ jointly reconstruct $\mathbf{U}_1$ and $\mathbf{v}_1$.
  3: $\mathcal{S}_i$ computes $[\mathbf{G}_1 \cdot \mathbf{p}]_i = i \cdot \mathbf{U}_1\mathbf{v}_1 + \mathbf{U}_1[\mathbf{y}_1]_i + [\mathbf{X}_1]_i\mathbf{v}_1 + [\mathbf{z}_1]_i$.
  4: $\mathcal{S}_i$ computes $[\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{b}_1]_i = [\mathbf{G}_1 \cdot \mathbf{p}]_i + [\mathbf{b}_1]_i$.
```

Fig. 3. Computing the secret shares of $\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{b}_1$.

$x < 0$, we need to compute two values $t_3 = (-1) \cdot (n_1x^2 + (-1) \cdot c_1x + d_1)$ and $t_4 = (-1) \cdot (n_2x^2 + (-1) \cdot c_2x + d_2)$, i.e., $t_3 = -n_1x^2 + c_1x - d_1$ and $t_4 = -n_2x^2 + c_2x - d_2$. The details of the secure evaluation of the $tanh(\cdot)^*$ function is given in Fig. 4.

Based on the above secure evaluation mechanism, we can enable the two cloud servers to obtain the secret shares of $tanh(\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{b}_1)^*$. Then, the two cloud servers continue to compute the secret shares of $\mathbf{G_2}tanh(\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{b}_1)^* + \mathbf{b}_2$. Suppose that we have the multiplication triplet $\mathbf{X}_2, \mathbf{y}_2, \mathbf{z}_2$, where $\mathbf{X}_2$ has the same dimension as $\mathbf{G}_2$, $\mathbf{y}_2$ has the same dimension as $tanh(\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{b}_1)^*$, and $\mathbf{z}_2 = \mathbf{X}_2 \cdot \mathbf{y}_2$. The computation of the secret shares of $\mathbf{G_2}tanh(\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{b}_1)^* + \mathbf{b}_2$ is given in Fig. 5.

Note that after each noisy patch is securely denoised and the secret shares of denoised patches are obtained, each cloud server can further obtain the secret share of the denoised image by placing the secret shares of denoised patches at the locations of their noisy counterparts and averaging/weighting the overlapping region, which just requires local computation of secure addition and multiplication by constants [16]. We denote the secret share of the denoised image held by each cloud server $\mathcal{S}_i$ as $[\mathbf{D}]_i$.

```
Input: Secret shares $[x]_i$ of an element $x$ in $\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{b}_1$, a secret-shared multiplication triplet $[a]_i$, $[b]_i$, $[c]_i$ ($c = a \cdot b$).
Output: $[tanh(x)^*]_i$.
  1: $\mathcal{S}_i$ first computes $[u]_i = [x]_i - [a]_i$ and $[v]_i = [x]_i - [b]_i$ locally.
  2: $\mathcal{S}_0$ and $\mathcal{S}_1$ jointly reconstruct $u$ and $v$.
  3: $\mathcal{S}_i$ computes the share $[x^2]_i = i \cdot u \cdot v + u \cdot [b]_i + v \cdot [a]_i + [c]_i$.
  4: $\mathcal{S}_i$ computes $[t_1]_i = n_1[x^2]_i + c_1[x]_i + i \cdot d_1$, $[t_2]_i = n_2[x^2]_i + c_2[x]_i + i \cdot d_2$, $[t_3]_i = -n_1[x^2]_i + c_1[x]_i - i \cdot d_1$, and $[t_4]_i = -n_2[x^2]_i + c_2[x]_i - i \cdot d_2$.
  5: $\mathcal{S}_1$ prepares a garbled circuit that takes as input the garbled values of the secret shares of $x, t_1, t_2, t_3, t_4$, and a random value $r$.
  6: $\mathcal{S}_1$ sends the garbled circuit, and the garbled values $\widetilde{[x]_1}$, $\widetilde{[t_1]_1}$, $\widetilde{[t_2]_1}$, $\widetilde{[t_3]_1}$, $\widetilde{[t_4]_1}$, and $\widetilde{r}$ to $\mathcal{S}_0$.
  7: $\mathcal{S}_0$ runs an oblivious transfer protocol with $\mathcal{S}_1$ to obtain its garbled values $\widetilde{[x]_0}$, $\widetilde{[t_1]_0}$, $\widetilde{[t_2]_0}$, $\widetilde{[t_3]_0}$, and $\widetilde{[t_4]_0}$.
  8: $\mathcal{S}_0$ evaluates the garbled circuit and obtains the output $[tanh(x)^*]_0 = tanh(x)^* - r$, while $\mathcal{S}_1$ holds $[tanh(x)^*]_1 = r$.
```

Fig. 4. Secure customized evaluation of the $tanh(\cdot)^*$ function.

```
Input: $[\mathbf{G}_2]_i$, $[tanh(\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{b}_1)^*]_i$, $[\mathbf{b}_2]_i$, $[\mathbf{X}_2]_i$, $[\mathbf{y}_2]_i$, and $[\mathbf{z}_2]_i$.
Output: $[\mathbf{G}_2 \cdot tanh(\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{b}_1)^* + \mathbf{b}_2]_i$.
  1: $\mathcal{S}_i$ computes $[\mathbf{U}_2]_i = [\mathbf{G}_2]_i - [\mathbf{X}_2]_i$ and $[\mathbf{v}_2]_i = [tanh(\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{b}_1)^*]_i - [\mathbf{y}_2]_i$.
  2: $\mathcal{S}_0$ and $\mathcal{S}_1$ jointly reconstruct $\mathbf{U}_2$ and $\mathbf{v}_2$.
  3: $\mathcal{S}_i$ computes $[\mathbf{G}_2 \cdot tanh(\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{b}_1)^*]_i = i \cdot \mathbf{U}_2\mathbf{v}_2 + \mathbf{U}_2[\mathbf{y}_2]_i + [\mathbf{X}_2]_i\mathbf{v}_2 + [\mathbf{z}_2]_i$.
  4: $\mathcal{S}_i$ computes $[\mathbf{G}_2 \cdot tanh(\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{b}_1)^* + \mathbf{b}_2]_i = [\mathbf{G}_2 \cdot tanh(\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{b}_1)^*]_i + [\mathbf{b}_2]_i$.
```

Fig. 5. Computing the secret shares of $\mathbf{G}_2 \cdot tanh(\mathbf{G}_1 \cdot \mathbf{p} + \mathbf{b}_1)^* + \mathbf{b}_2$.

4.3 Image Decryption

Upon receiving the request from the receiver for the denoised image $\mathbf{D}$, each cloud server $\mathcal{S}_i$ sends $[\mathbf{D}]_i$ to the receiver. Then, the receiver can recover the denoised image by combining the two secret shares, i.e., $\mathbf{D} = [\mathbf{D}]_0 + [\mathbf{D}]_1$.

4.4 Security Guarantees

We summarize the security guarantees of our privacy-preserving DNN based image denoising design by the following theorem.

Theorem 1. *Given that the two cloud servers are honest-but-curious adversaries and do not collude with each other, our design ensures that the two cloud servers learn nothing about the image content (*$\mathbf{I}$ *or* $\mathbf{D}$*), and the values in the private parameters* $\mathbf{G}_j$ *and* $\mathbf{b}_j$ *of the DNN model.*

Proof. Given the security of the underlying cryptographic techniques of additive secret sharing, Beaver's multiplication technique, and garbled circuits, the security guarantees can be proved immediately according to the modular sequential theorem [61], [62], by showing that each step of our design is secure.

First, before encrypted image denoising starts, the two cloud servers receive the secret shares of $\mathbf{G}_j$ and $\mathbf{b}_j$ of the DNN model and of each image $\mathbf{I}$. The security of additive secret sharing ensures that each cloud server learns nothing about the plaintext values underlying its secret shares.

Subsequently, during the procedure of encrypted DNN based denoising, the two cloud servers operate over their respective secret shares locally and also have some interactions with each other, either via the Beaver's technique for secure multiplication or via garbled circuits for secure evaluation of the (approximated) activation function. The security of the Beaver's technique ensures that each cloud server learns nothing from the interaction for secure multiplication. Regarding the execution of garbled circuits, the security of garbled circuits, which is formally proved in [55], ensures that nothing is disclosed to the two cloud servers except the output produced on the $\mathcal{S}_0$ side. Recall that the output from garbled circuit evaluation is a secret share of the true evaluation result on the $\mathcal{S}_0$ side, where the other share is kept by $\mathcal{S}_1$. So, each cloud server also learns nothing from the execution of garbled circuits. In a nutshell, each cloud server only observes "random-looking" values and thus learns nothing. □

Authorized licensed use limited to: POLO BIBLIOTECARIO DI INGEGNERIA. Downloaded on June 02,2021 at 15:51:59 UTC from IEEE Xplore. Restrictions apply.

```
Input: Secret shares $[v^*]_i$ of a value $v^*$ that is scaled by $2^{2s}$.
Output: Secret shares $[v]_i$ of a value $v$ where $v = \lfloor v^*/2^s \rceil$.
1: $\mathcal{S}_0$ computes $v'_r = [v^*]_0 + r$ ($r \xleftarrow{\$} \mathbb{Z}_{2^{l+u}}$), and sends it to $\mathcal{S}_1$.
2: $\mathcal{S}_1$ computes $v^*_r = [v^*]_0 + r + [v^*]_1$, and compute $[v]_1 = v^*_r \gg s \bmod 2^{l-s}$.
3: $\mathcal{S}_0$ sets its share of $v$ to $[v]_0 = -(r \gg s) \bmod 2^{l-s}$.
```

Fig. 6. The secure rescaling protocol used in our security design. Note that there is no modular arithmetic in computing v'_r and v^*_r.

4.5 Our Practical Considerations

With the design just described above as a promising foundation, we now further explore and address some practical considerations for deploying our proposed privacy-preserving DNN based image denoising services.

Number Representation. Our above security design works in the ring $\mathbb{Z}_{2^l}$. However, plaintext DNN based image denoising requires computation over real values. So, we need to consider how to properly represent the values for computation in secure DNN based image denoising. For efficiency consideration, we follow most of prior works (e.g., [27], [30], [63], to just list a few) on secure computation and resort to fixed-point representation for any non-integers. The intuitive idea is to scale and round non-integers to integers. In particular, given a real value v, its fixed-point representation is $\overline{v} = \lfloor v \cdot 2^s \rfloor$, where the fixed scaling factor s controls the precision. Then, we can map $\overline{v}$ to the ring $\mathbb{Z}_{2^l}$ by computing $\overline{v} \bmod 2^l$.

Secure Shared-Value Rescaling. Note that after transforming the inputs for secure computation, all intermediate results are also expressed with fixed-point representation. Therefore, for computation correctness, we need to make ensure that any intermediate values should not exceed the bit length l. A noteworthy fact is that the multiplication of two scaled values would lead to a scaling factor 2^{2s} for the actual result, which may quickly exceed the big-length l and cause overflow. Therefore, we consider how to scale down values scaled by 2^{2s}, before any subsequent multiplications take place. To this end, we resort to a very recent secure, effective, and efficient rescaling protocol in [63], which is given in Fig. 6. The secure rescaling protocol enables the two cloud servers holding secret shares of a value v^* scaled by 2^{2s} to obtain the secret shares of a value v scaled by 2^s, while learning nothing.

Local Pre-Processing and Post-Processing. While in principle we can put all the workload of denoising on the cloud side, this, however, may not be the best strategy from a practical perspective. In fact, DNN based image denoising requires data pre-processing and post-processing as described below [64]. For pre-processing, each original patch in the (noisy) image should be properly processed before being fed into the DNN model for denoising. Suppose that the noise level of the noisy image is σ and the noise level of the trained neural network model is σ^*. The general pre-processing for an original patch $\hat{\mathbf{p}}$ is as follows:

1). Scaling the noise: $\mathbf{p}' = \sigma^* \cdot \sigma^{-1} \cdot \hat{\mathbf{p}}$;
2). Normalizing the patch: $\tilde{\mathbf{p}} = (\mathbf{p}'/255 - 0.5) \cdot 5$;
3). Computing the mean shift: $m = \sum_{w=1}^{W} \tilde{\mathbf{p}}(w)/W + 0.5$, where $\tilde{\mathbf{p}}(w)$ is the w-th element in the patch $\tilde{\mathbf{p}}$, and W is the total number of elements;
4). Obtaining the actual neural network input: $\mathbf{p} = \tilde{\mathbf{p}} - m$.

In addition, after the DNN based denoising procedure, some post-processing is also required for each denoised patch $\mathbf{q}$, which is as follows:

1). Shifting the denoised patch: $\hat{\mathbf{q}} = \mathbf{q} + m$;
2). Reverse the normalization: $\mathbf{q}^* = (\hat{\mathbf{q}}/5 + 0.5) \cdot 255 \cdot (\sigma^* \cdot \sigma^{-1})^{-1}$.

Note that if $\sigma^* = \sigma$, only Step 2 is needed in pre-processing and the same applies to post-processing. As shown, the pre-processing and post-processing of patches takes a number of steps that require additions and multiplications. While such pre-processing and post-processing can be trivially done in the plaintext domain, handling them in the ciphertext domain would otherwise largely complicate the encrypted denoising procedure, and cause practically unnecessary performance overhead. Instead of doing such pre-processing and post-processing in the encrypted domain on the cloud side, we add slight modification to our above design by adopting the strategy of local pre-processing on the IoT gateway side and post-processing on the receiver side. That is, the IoT gateway sends secret shares of pre-processed patches to the cloud servers for denoising. Upon request, the cloud servers send the secret shares of the denoised patches to the receiver, who then combines the secret shares to recover the denoised patches and further performs all other post-processing to recover the whole denoised image. In this way, we avoid unnecessary computation/communication overhead in the ciphertext domain due to pre-processing and post-processing.

Reducing Communication Cost. The communication cost of the IoT gateway depends on the transmission of the secret shares to each cloud server. Hence, to reduce the communication cost of the IoT gateway, we can consider how to reduce the amount of secret shares to be transmitted by the IoT gateway. Inspired by prior work [36], our idea is to apply a keyed pseudo-random function (PRF) at the IoT gateway to generate the secret shares (i.e., random values) which are to be received by the cloud server $\mathcal{S}_1$. In particular, for each (pre-processed) patch $\mathbf{p}$, the secret share $[\mathbf{p}]_1$ is generated via $[\mathbf{p}]_1 = \{PRF_K(w)\}_{w=1}^{W}$, where K is a freshly generated PRF key for that patch; the secret share $[\mathbf{p}]_0$ is then derived as $[\mathbf{p}]_0 = \mathbf{p} - [\mathbf{p}]_1$. In this way, it is easy to see that the IoT gateway does not need to directly send the secret share $[\mathbf{p}]_1$ (which could contain a large number of elements) to the cloud server $\mathcal{S}_1$, and can just share the freshly generated key K to enable the cloud server $\mathcal{S}_1$ to reconstruct the share $[\mathbf{p}]_1$. Note that the security of PRF function ensures that the secret shares generated are indistinguishable from truly random values.

In addition, we remark that it is feasible to support batch input in our security design, and this can help reduce the communication cost on the cloud side due to encrypted multiplication. Recall that as introduced in Section 3, the input to the DNN model in DNN-based image denoising is a vectorized image patch [16]. So, the support for bath input will work by taking multiple image patches as input. To support batch input in our design, we just need to pack multiple input vectors (image patches) into an input matrix, and change from matrix-vector multiplication to matrix-matrix multiplication during the secure DNN computation procedure. Suppose that a DNN weight matrix has a size $m \times n$, and the

Authorized licensed use limited to: POLO BIBLIOTECARIO DI INGEGNERIA. Downloaded on June 02,2021 at 15:51:59 UTC from IEEE Xplore. Restrictions apply.

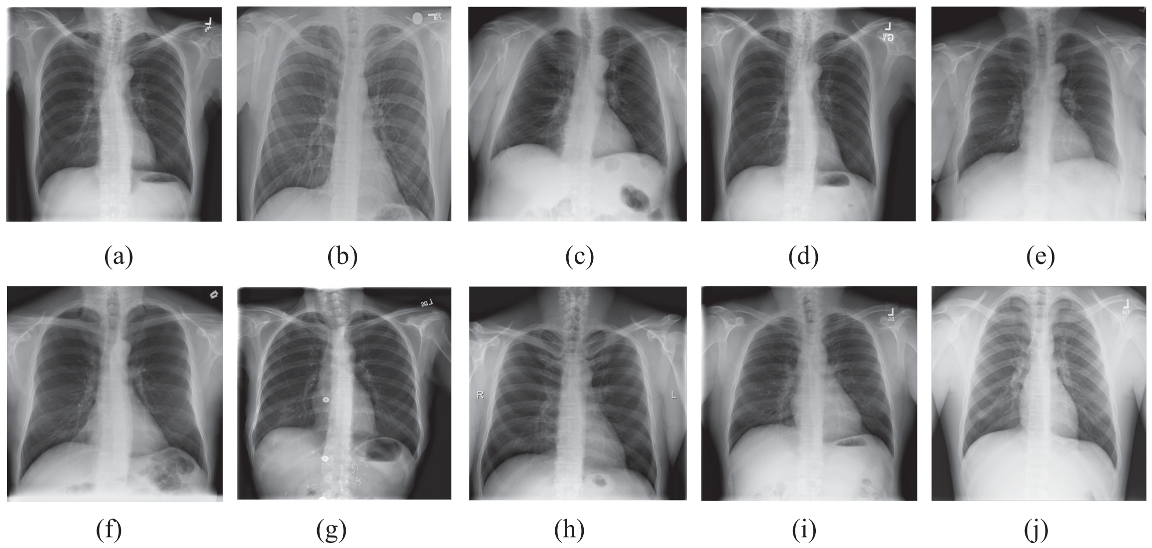


Fig. 7. Examples of the test images used in our experiments. They are denoted by “a”, “b”, $\cdots$, “j” for convenience.

batch size is p. With the typical ring size $l = 64$, the communication cost in the batch-input case due to encrypted multiplication would be $16 \times (m \times n + n \times p)$ bytes, as opposed to $16 \times p \times (m \times n + n)$ bytes in the plain case. So, the communication cost reduction in this case would be $16 \times mn \times (p-1)$ bytes. We point out that the usage of batch input might also lead to improvement in runtime cost, through implementation-wise tricks in optimizing matrix-matrix multiplication. However, this is orthogonal to our security design and out of the scope of this paper.

5 Experiments

5.1 Implementation

We implement a proof-of-concept prototype for our security design in C++. We use GMP (v6.1.2) for calculations over ring, and Eigen (v3.3.4) for operations on matrix. To reduce the overhead of dynamic memory management during denoising, we associate the neural network model with preallocated buffers that are reusable for intermediate results. We also implement a multi-threaded version, where a pool of worker threads can denoise distinct patches in parallel with their own buffers. For garbled circuits, we use the GASH-lang,[1] a programming framework for Yao’s garbled circuit. GASH-lang includes a compiler of its own language, and an implementation of Yao’s garbled circuit protocol that incorporates recent optimizations such as Free-XOR [65], fixed key block cipher [66], and garbled row reduction [67]. It enables writing high-level function using a circuit description syntax, which is similar to C. We deploy and evaluate the prototype on Microsoft Azure Standard F64s v2 instance, as well as a local client with Intel E3-1505 v5 CPU (2.8 GHz) and 16 GB memory. To ensure computation accuracy, we need to use sufficiently large ring and scaling factor [59]. In our experiments the ring size 2^l is 2^{64} and the scaling factor 2^s is 2^{18}. The statistical security parameter u in the secure rescaling protocol is set to 40 [68].

For demonstration, we use a real healthcare-related dataset ChestX-ray8 [69] in our experiments. We form our test dataset by randomly selecting 50 images from ChestX-ray8. In our experiments, each test image has a size 512×512, and some examples are provided in Fig. 7. Following existing works on plaintext/secure image denoising [16], [32], [33], [46], [70], we generate noisy images by adding zero-mean Gaussian noise with standard deviations ranging from $\sigma = 15$ to $\sigma = 35$ to the test images. Here, we note that the noises in many medical images may be Poisson noises in the first place. However, as indicated by existing works [70], [71], the Poisson noises in medical images can be transformed into the universal Gaussian noises through the variance stability transform (VST), and Gaussian denoising approaches can then effectively. We also have the observation that due to these facts, experimenting with medical image denoising under Gaussian noises also appeared in the plaintext domain [47], [70]. So, we consider our practice to be consistent with plaintext works and we stress that this is effectively orthogonal to our security design.

We use the trained DNN model[2] in the plaintext work [72] for test. This DNN model is trained from the ImageNet dataset [73]. It consists of four large hidden layers and has a large size. The sizes of the hidden layers are 3072, 3072, 2559, and 2047, respectively. The output layer has a size 289, i.e., a denoised patch with size 17×17. The noise level parameter σ^* of the model is 25. For denoising, each test noisy image is decomposed into overlapping patches of size 17×17 with a stride size 3. To denoise a noisy patch $\mathbf{p}$, the neural network model takes as input a noisy patch $\mathbf{p}^*$ of size 39×39, which includes not only the corresponding noisy pixels of $\mathbf{p}$ but also the surrounding ones. We will use two widely adopted metrics, namely peak signal to noise ratio (PSNR) and structural similarity (SSIM) [74], to measure the objective denoising quality. The PSNR metric measures the intensity difference between two images, while the SSIM measures the perceptual quality difference.

5.2 Performance Evaluation

5.2.1 Denoising Quality

We first conduct a denoising quality evaluation to validate the effectiveness of our security design. We make comparison

1. https://github.com/xiaottang2/gash

2. At http://people.tuebingen.mpg.de/burger/neural_denoising/

Authorized licensed use limited to: POLO BIBLIOTECARIO DI INGEGNERIA. Downloaded on June 02,2021 at 15:51:59 UTC from IEEE Xplore. Restrictions apply.

TABLE 2
Comparison of PSNR (in dB) and SSIM Results for 10 Example Test Images Shown in Fig. 7

Image	Method	$\sigma = 15$		$\sigma = 20$		$\sigma = 25$		$\sigma = 30$		$\sigma = 35$	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
a	Plaintext	36.13	0.92	34.85	0.9	34.07	0.88	32.51	0.86	31.31	0.85
	Secure	**35.82**	**0.91**	**34.55**	**0.89**	**33.73**	**0.88**	**32.26**	**0.86**	**31.06**	**0.85**
b	Plaintext	37.21	0.95	35.95	0.93	34.95	0.91	34.23	0.9	33.62	0.89
	Secure	**37.15**	**0.95**	**35.9**	**0.93**	**34.78**	**0.91**	**34.19**	**0.9**	**33.58**	**0.89**
c	Plaintext	37.29	0.94	35.85	0.93	34.96	0.91	33.47	0.9	32.51	0.89
	Secure	**37.12**	**0.94**	**35.72**	**0.93**	**34.63**	**0.91**	**33.3**	**0.9**	**32.35**	**0.89**
d	Plaintext	36.16	0.93	34.96	0.9	34.24	0.89	32.64	0.87	31.6	0.86
	Secure	**35.85**	**0.92**	**34.67**	**0.9**	**33.89**	**0.89**	**32.37**	**0.87**	**31.34**	**0.86**
e	Plaintext	37.4	0.93	36.08	0.92	35.3	0.91	33.8	0.9	32.81	0.89
	Secure	**37.26**	**0.93**	**35.94**	**0.92**	**34.93**	**0.91**	**33.68**	**0.9**	**32.7**	**0.89**
f	Plaintext	36.62	0.95	35.82	0.93	34.99	0.92	33.71	0.91	32.86	0.9
	Secure	**36.42**	**0.95**	**35.59**	**0.93**	**34.62**	**0.92**	**33.5**	**0.91**	**32.65**	**0.9**
g	Plaintext	35.26	0.93	34.3	0.91	33.58	0.89	31.91	0.87	30.91	0.85
	Secure	**34.9**	**0.93**	**33.95**	**0.91**	**33.23**	**0.89**	**31.58**	**0.86**	**30.59**	**0.85**
h	Plaintext	36.6	0.94	35.18	0.92	34.34	0.9	32.82	0.89	31.78	0.87
	Secure	**36.4**	**0.94**	**35**	**0.92**	**34.05**	**0.9**	**32.6**	**0.88**	**31.55**	**0.87**
i	Plaintext	36.97	0.92	35.44	0.91	34.66	0.9	32.86	0.88	31.83	0.87
	Secure	**36.75**	**0.92**	**35.21**	**0.9**	**34.36**	**0.89**	**32.67**	**0.88**	**31.64**	**0.87**
j	Plaintext	38.36	0.97	36.81	0.96	35.5	0.94	33.67	0.94	32.38	0.93
	Secure	**38.25**	**0.97**	**36.71**	**0.96**	**35.1**	**0.94**	**33.6**	**0.93**	**32.35**	**0.92**

TABLE 3
Comparison of Average PSNR (in dB) and SSIM Results of 50 Test Images

Image	Method	$\sigma = 15$		$\sigma = 20$		$\sigma = 25$		$\sigma = 30$		$\sigma = 35$	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Ave.	Plaintext	36.18	0.94	35.36	0.92	34.61	0.91	33.05	0.89	31.96	0.88
	Secure	**35.94**	**0.94**	**35.11**	**0.92**	**34.26**	**0.91**	**32.8**	**0.89**	**31.71**	**0.87**

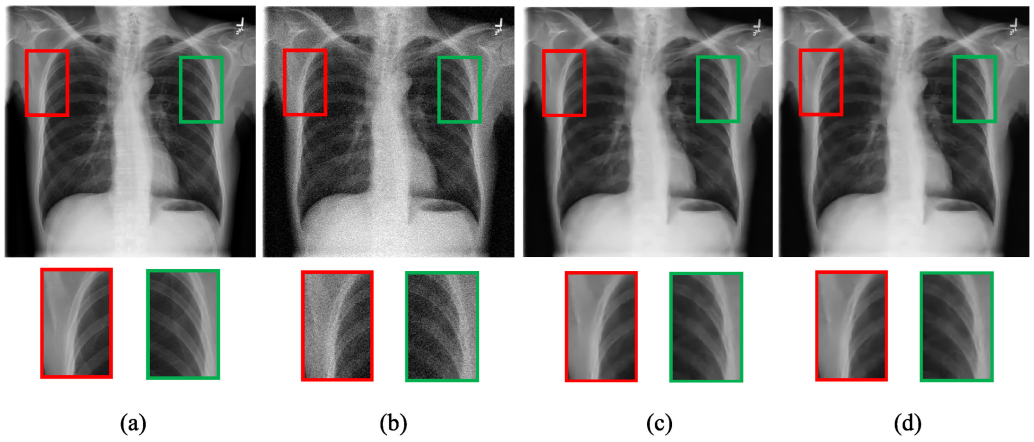


Fig. 8. Example visual denoising result for one test image (Fig. 7a) when $\sigma = 15$. (a) Original image; (b) Noisy image; (c) Plaintext result; (d) Our result.

with the plaintext method underlying our security design, which serves as the baseline and uses the original $tanh(\cdot)$ function. First, we conduct an objective evaluation and report the results. Table 2 provides the objective PSNR and SSIM results for the 10 example test images displayed in Fig. 7. In Table 3, we further give the average PSNR and SSIM results for our experiments on 50 test images. The results reveal that the objective denoising quality achieved by our security design is comparable to that by the plaintext method. For example, when $\sigma = 20$, the PSNR of our security design is just 0.25 dB lower than that of the plaintext method, while the SSIM of our security design is the same as in the plaintext method. On

Authorized licensed use limited to: POLO BIBLIOTECARIO DI INGEGNERIA. Downloaded on June 02,2021 at 15:51:59 UTC from IEEE Xplore. Restrictions apply.

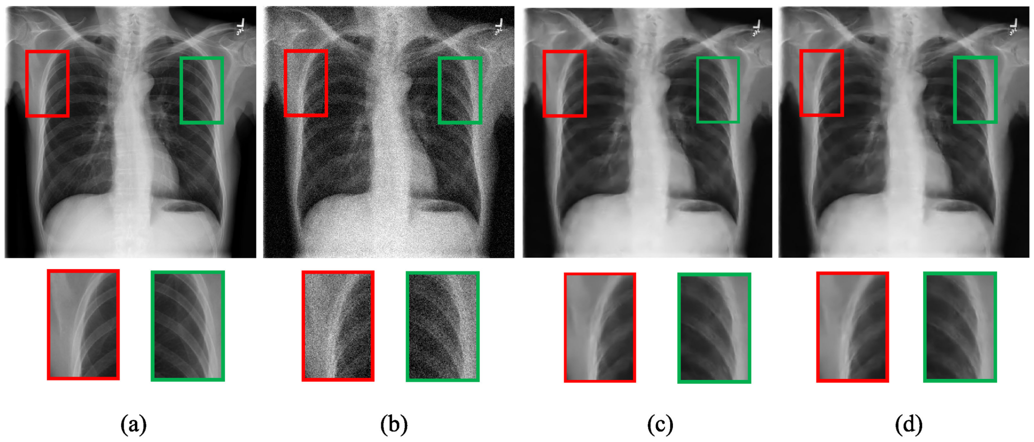


Fig. 9. Example visual denoising result for one test image (Fig. 7a) when $\sigma = 20$. (a) Original image; (b) Noisy image; (c) Plaintext result; (d) Our result.

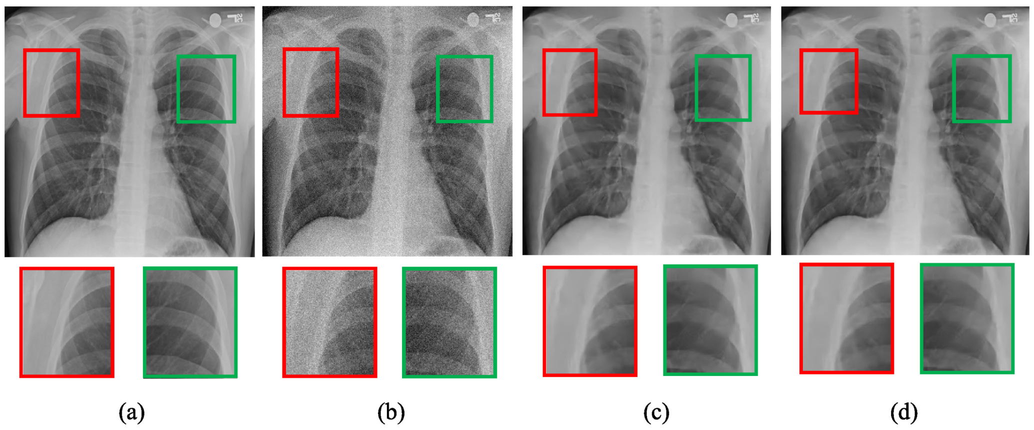


Fig. 10. Example visual denoising result for one test image (Fig. 7b) when $\sigma = 15$. (a) Original image; (b) Noisy image; (c) Plaintext result; (d) Our result.

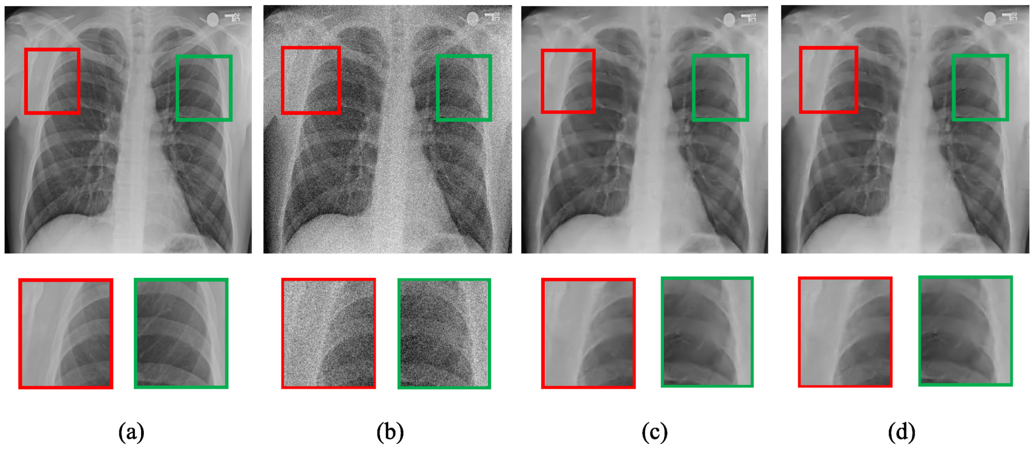


Fig. 11. Example visual denoising result for one test image (Fig. 7b) when $\sigma = 20$. (a) Original image; (b) Noisy image; (c) Plaintext result; (d) Our result.

average, the PSNR of our security design is only 0.27 dB lower than the plaintext method, while the SSIM is only 0.002 lower.

In addition to an objective denoising quality evaluation, we also conduct a visual denoising quality evaluation. Figs. 8, 9, 10, and 11 provide some example visual denoising results. It can be observed that the overall visual denoising result produced by our security design is also comparable to the plaintext method. To show the visual details clearly, we highlight two cropped regions with a red rectangle and a green rectangle respectively, for each image displayed in Figs. 8, 9, 10, and 11. The two cropped regions are displayed at the bottom of each corresponding image. We can observe that the visual details produced by our security design and the plaintext method are highly close.

Authorized licensed use limited to: POLO BIBLIOTECARIO DI INGEGNERIA. Downloaded on June 02,2021 at 15:51:59 UTC from IEEE Xplore. Restrictions apply.

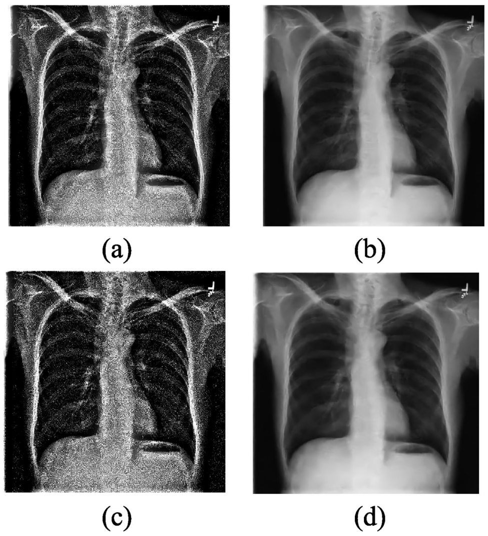


Fig. 12. Comparison of denoising results from plaintext denoising via activation function approximation from [41] (hereafter referred to briefly as [41] approx. result.) and from our security design. (a) [41] approx. result (for $\sigma = 15$), with PSNR 14.56 dB; (b) Our result (for $\sigma = 15$), with PSNR 35.82 dB; (c) [41] approx. result (for $\sigma = 20$), with PSNR 12.34 dB; (d) Our result (for $\sigma = 20$), with PSNR 34.55 dB.

We note that the work in [41] builds on a determined and closed-form approximation of the Sigmoid function to propose a security design for privacy-preserving image classification. Considering that there is a relationship (constant rescaling and shifting) between the sigmoid function and the $tanh(\cdot)$ function, for completeness we now evaluate the denoising quality of plaintext denoising via approximated $tanh(\cdot)$ function derived from the approximated sigmoid function used in [41]. And we make comparison with results from our security design. The denoising results, as shown in Fig. 12, indicate that the approximation design in [41] which originally targets *image classification* performs not well and is not suitable for image denoising. So, the security design of [41] is not applicable to our target image denoising problem.

5.2.2 Computation Performance

We now turn to the computation performance of our security design. First, we examine the computation cost the IoT gateway side. Recall that the computation cost at the IoT gateway side includes the pre-processing of patches and the encryption of patches (i.e., generation of secret shares). Our encryption is highly efficient: only 0.145 ms per patch; and the full pre-processing (i.e., 4 steps as presented in 4.5) takes 7.451 μs per patch. Note that the overall computation cost for an image inherently depends on the image size (deciding number of patches). As the encryption mechanism is highly efficient, it just takes 3.978 s to process a test image at the IoT gateway side.

We then measure the computation performance at the cloud side. First, we measure the computation cost of each atomic secure operation to examine the efficiency. Table 4 gives the computation cost of each secure atomic operation, including secure addition, secure multiplication, secure evaluation of $tanh(\cdot)^*$, and secure rescaling. It is shown that these operations are highly efficient. For a single encrypted DNN computation, we note that the overall cost is decided by the size of the DNN model. Given such a large-size DNN model (which requires $> 3 \cdot 10^7$ multiplications and $> 10^4$ evaluations of the activation function) used in our experiments, it takes around 111.4 minutes to complete a single encrypted DNN computation (i.e., denoise a patch) on the cloud side, which is practically affordable. We note that this computation cost is dominated by the garbled circuit online execution for secure evaluation of the $tanh(\cdot)^*$ function, and it can be effectively brought down by parallelizing secure $tanh(\cdot)^*$ evaluation of different values at each layer. Additionally, it is worth noting that the denoising of different encrypted patches are independent and allows parallelism which can be effectively accomplished on the resource-rich and economical cloud.

We also evaluate the computation cost at the receiver side to obtain a denoised image. Recall that the computation cost includes the decryption of ciphertexts of denoised patches (i.e., combination of patch shares), the post-processing of the denoised patches, and the recovery of the whole image. The decryption of ciphertexts of each denoised patch only takes 0.087 ms, which is highly efficient. The post-processing of each denoised patch only takes 8.818 μs, and the recovery of the whole image takes 2.443 s.

5.2.3 Communication Performance

We now examine the communication performance of our security design. First of all, we report the communication cost of the IoT gateway. Recall that for each image, the IoT gateway needs to send the shares of each patch to the cloud. With two shares of each patch, the communication cost per patch is 23.77 KB. When using the PRF-based optimization trick proposed in 4.5, we can reduce the communication cost per patch to just 11.91 KB. For the communication performance on the cloud side for encrypted denoising, we first report the communication cost for each secure atomic operation that requires interactions between the two cloud servers. Such operations include secure multiplication, secure evaluation of $tanh(\cdot)^*$, and secure rescaling. The communication cost between the two cloud servers for atomic secure multiplication and secure rescaling are just 256 bits and 104 bits respectively. For the secure evaluation of $tanh(\cdot)^*$, the communication cost between the two cloud servers is about 74.5 KB, which is dominated by the garbled circuit online execution. Overall, the communication cost between the two cloud servers (i.e., communication cost) for denoising

TABLE 4
Computation Cost of Secure Atomic Operations in Our Security Design

IoT gateway	Cloud				Receiver
Patch shares gen.	Sec. add.	Sec. mul.	Sec. eval.	Sec. rescal.	Patch shares comb.
0.145 ms	0.071 μs	0.556 μs	0.619 s	0.164 μs	0.087 ms

Authorized licensed use limited to: POLO BIBLIOTECARIO DI INGEGNERIA. Downloaded on June 02,2021 at 15:51:59 UTC from IEEE Xplore. Restrictions apply.

one encrypted patch is around 1,207 MB, which is practically affordable on the cloud side. Regarding the receiver, the communication cost is due to obtaining the shares of denoised patches from the two cloud servers, and the per-patch cost is only 4.52 KB.

6 Conclusion and Future Work

In this paper, we presented the first secure system framework enabling privacy-preserving DNN based image denoising services on the cloud. In our security design, we uniquely bridged together lightweight cryptographic techniques like additive secret sharing and garbled circuits to seamlessly embrace the operations required by DNN based image denoising. We explored and constructed secure and efficient mechanisms customized for the atomic operations underlying DNN based image denoising, like the secure evaluation of the non-linear activation function. Our security design ensures that the private image content and DNN model are all kept private throughout the whole cloud-based service flow. Extensive experiments were conducted, and the results show that our security design achieves denoising quality comparable to that in the plaintext domain, and is highly cost-efficient on the local side and practically affordable on the cloud side.

For future directions, it would be interesting to extend our initial security design to support other types of DNN models for image denoising such as deep convolutional neural networks [29], [75] that have more complicated structures, as well as to explore more efficient and effective activation function approximation techniques for further accelerating secure DNN based image denoising.

Acknowledgments

This work was supported in part by the Research Grants Council of Hong Kong under Grant CityU 11276816, Grant CityU 11212717, and Grant CityU C1008-16G, the Innovation and Technology Commission of Hong Kong under ITF Project ITS/168/17, the National Natural Science Foundation of China under Grant 61572412, the Macau Science and Technology Development Fund under Grant FDCT/022/2017/A1 and Grant FDCT/077/2018/A2, the Research Committee at the University of Macau under Grant MYRG2016-00137-FST and Grant MYRG2018-00029-FST, and a Microsoft Azure grant for research.

References

[1] A. Briassouli, J. Benois-Pineau, and A. G. Hauptmann, "Overview of multimedia in healthcare," in *Health Monitoring and Personalized Feedback using Multimedia Data*. New York, NY, USA: Springer, 2015, pp. 1–6.
[2] N. M. Daher, "The internet of medical imaging things is here," 2016. [Online]. Available: http://www.diagnosticimaging.com/pacs-and-informatics/internet-medical-imaging-things-here
[3] R. Mcaskill, "Remote monitoring pushing healthcare IoT trends," 2016. [Online]. Available: https://mhealthintelligence.com/news/remote-monitoring-pushing-healthcare-iot-trends
[4] R. Bachrach, "Handheld imaging devices could transform health care," 2015. [Online]. Available: http://viztek.net/blog/handheld-imaging-devices-could-transform-health-care/
[5] D. Smith, "Butterfly network developing handheld MRI & ultrasound scanner," 2014. [Online]. Available: https://www.partspeople.com/blog/2014/11/06/butterfly-network-developing-handheld-mri-ultrasound-scanner/
[6] X. Yi, F. Rao, E. Bertino, and A. Bouguettaya, "Privacy-preserving association rule mining in cloud computing," in *Proc. 10th ACM Symp. Inf. Comput. Commun. Secur.*, 2015, pp. 493–450.
[7] H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, and X. S. Shen, "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data," *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 3, pp. 312–325, May/Jun. 2016.
[8] C. Zhang, L. Zhu, C. Xu, and R. Lu, "PPDP: An efficient and privacy-preserving disease prediction scheme in cloud-based e-healthcare system," *Future Generation Comp. Syst.*, vol. 79, pp. 16–25, 2018.
[9] Y. Zhang, H. Huang, Y. Xiang, L. Y. Zhang, and X. He, "Harnessing the hybrid cloud for secure big image data service," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1380–1388, Oct. 2017.
[10] Y. Zheng, X. Yuan, X. Wang, J. Jiang, C. Wang, and X. Gui, "Toward encrypted cloud media center with secure deduplication," *IEEE Trans. Multimedia*, vol. 19, no. 2, pp. 251–265, Feb. 2017.
[11] X. Liu, D. Zhai, D. Zhao, G. Zhai, and W. Gao, "Progressive image denoising through hybrid graph laplacian regularization: A unified framework," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1491–1503, Apr. 2014.
[12] M. L. L. De Faria, C. E. Cugnasca, and J. R. A. Amazonas, "Insights into iot data and an innovative dwt-based technique to denoise sensor signals," *IEEE Sensors J.*, vol. 18, no. 1, pp. 237–247, Jan. 2018.
[13] P. Chatterjee and P. Milanfar, "Patch-based near-optimal image denoising," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1635–1649, Apr. 2012.
[14] A. Wong, A. K. Mishra, W. Zhang, P. W. Fieguth, and D. A. Clausi, "Stochastic image denoising based on markov-chain monte carlo sampling," *Signal Process.*, vol. 91, no. 8, pp. 2112–2120, 2011.
[15] C. Sutour, C. Deledalle, and J. Aujol, "Adaptive regularization of the nl-means: Application to image and video denoising," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3506–3521, Aug. 2014.
[16] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with bm3d?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2392–2399.
[17] S. Zhang, X. Chen, J. Wang, Z. Zhan, and J. Li, "Verifiable privacy-preserving single-layer perceptron training scheme in cloud computing," *Soft Comput.*, vol. 22, pp. 7719–7732, 2018.
[18] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proc. 41st Annu. ACM Symp. Theory Comput.*, 2009, pp. 169–178.
[19] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) LWE," in *Proc. 52nd Annu. Symp. Found. Comput. Sci.*, 2011, pp. 97–106.
[20] R. K. H. Tai, J. P. K. Ma, Y. Zhao, and S. S. M. Chow, "Privacy-preserving decision trees evaluation via linear functions," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2017, pp. 494–512.
[21] P. Li, T. Li, H. Ye, J. Li, X. Chen, and Y. Xiang, "Privacy-preserving machine learning with multiple data providers," *Future Generation Comput. Syst.*, vol. 87, pp. 341–350, 2018.
[22] Y. Zheng, H. Duan, and C. Wang, "Learning the truth privately and confidently: Encrypted confidence-aware truth discovery in mobile crowdsensing," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 10, pp. 2475–2489, Oct. 2018.
[23] C. Cai, Y. Zheng, and C. Wang, "Leveraging crowdsensed data streams to discover and sell knowledge: A secure and efficient realization," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst.*, 2018, pp. 589–599.
[24] S. Hu, M. Li, Q. Wang, S. S. M. Chow, and M. Du, "Outsourced biometric identification with privacy," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 10, pp. 2448–2463, Oct. 2018.
[25] Y. Zheng, H. Duan, X. Yuan, and C. Wang, "Privacy-aware and efficient mobile crowdsensing with truth discovery," *IEEE Trans. Dependable Secure Comput.*, vol. 99, no. 99, pp. 1–1, Jun. 2017, doi: 10.1109/TDSC.2017.2 753 245.
[26] T. Gupta, H. Fingler, L. Alvisi, and M. Walfish, "Pretzel: Email encryption and provider-supplied functions are compatible," in *Proc. Confe ACM Spec. Interest Group Data Commun.*, 2017, pp. 168–182.
[27] V. Nikolaenko, U. Weinsberg, S. Ioannidis, M. Joye, D. Boneh, and N. Taft, "Privacy-preserving ridge regression on hundreds of millions of records," in *Proc. IEEE Symp. Secur. Privacy*, 2013, pp. 334–348.
[28] F. Baldimtsi, D. Papadopoulos, S. Papadopoulos, A. Scafuro, and N. Triandopoulos, "Server-aided secure computation with off-line parties," in *Proc. Eur. Symp. Res. Comput. Secur.*, 2017, pp. 103–123.
[29] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

Authorized licensed use limited to: POLO BIBLIOTECARIO DI INGEGNERIA. Downloaded on June 02,2021 at 15:51:59 UTC from IEEE Xplore. Restrictions apply.

[30] J. Liu, M. Juuti, Y. Lu, and N. Asokan, "Oblivious neural network predictions via minionn transformations," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 619–631.

[31] C. Lin and J. Wang, "A digital circuit design of hyperbolic tangent sigmoid function for neural networks," in *Proc. IEEE Int. Symp. Circuits Syst.*, 2008, pp. 856–859.

[32] Y. Zheng, H. Cui, C. Wang, and J. Zhou, "Privacy-preserving image denoising from external cloud databases," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 6, pp. 1285–1298, Jun. 2017.

[33] X. Hu, W. Zhang, K. Li, H. Hu, and N. Yu, "Secure nonlocal denoising in outsourced images," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 3, pp. 40:1–40:23, 2016.

[34] Y. Zheng, C. Wang, and J. Zhou, "Toward secure image denoising: A machine learning based realization," *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2018, pp. 6936–6940.

[35] R. Gilad-Bachrach, N. Dowlin, K. Laine, K. E. Lauter, M. Naehrig, and J. Wernsing, "Cryptonets: Applying neural networks to encrypted data with high throughput and accuracy," in *Proc. 33rd Int. Conf. Int. Conf. Mach. Learn. - Vol. 48*, 2016, pp. 201–210.

[36] M. S. Riazi, C. Weinert, O. Tkachenko, E. M. Songhori, T. Schneider, and F. Koushanfar, "Chameleon: A hybrid secure computation framework for machine learning applications," in *Proc. Asia Conf. Comput. Commun. Secur.*, 2018, pp. 707–721.

[37] B. D. Rouhani, M. S. Riazi, and F. Koushanfar, "Deepsecure: Scalable provably-secure deep learning," in *Proc. 55th Annu. Des. Autom. Conf.*, 2017, Art. no. 2.

[38] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan, "Gazelle: A low latency framework for secure neural network inference," *CoRR*, vol. abs/1801.05507, pp. 1–17, 2018.

[39] N. Chandran, D. Gupta, A. Rastogi, R. Sharma, and S. Tripathi, "Ezpc: Programmable, efficient, and scalable secure two-party computation," *IACR Cryptology ePrint Archive*, vol. 2017, 2017, Art. no. 1109.

[40] E. Hesamifard, H. Takabi, M. Ghasemi, and R. N. Wright, "Privacy-preserving machine learning as a service," *PoPETs*, vol. 2018, no. 3, pp. 123–142, 2018.

[41] P. Mohassel and Y. Zhang, "Secureml: A system for scalable privacy-preserving machine learning," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 19–38.

[42] L. D. Xu, W. He, and S. Li, "Internet of things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[43] J. Treadway, "Using an IoT gateway to connect the "Things" to the cloud," 2016. [Online]. Available: http://internetofthingsagenda.techtarget.com/feature/Using-an-IoT-gateway-to-connect-the-Things-to-the-cloud

[44] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the internet of things: Security and privacy issues," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 34–42, Mar./Apr. 2017.

[45] K. Xue, S. Li, J. Hong, Y. Xue, N. Yu, and P. Hong, "Two-cloud secure database for numeric-related SQL range queries with privacy preserving," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 7, pp. 1596–1608, Jul. 2017.

[46] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[47] D. H. Trinh, M. Luong, F. Dibos, J. Rocchisani, C. D. Pham, and T. Q. Nguyen, "Novel example-based method for super-resolution and denoising of medical images," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1882–1895, Apr. 2014.

[48] E. Luo, S. H. Chan, and T. Q. Nguyen, "Adaptive image denoising by targeted databases," *IEEE Trans. Image Process.*, vol. 24, no. 7, pp. 2167–2181, Jul. 2015.

[49] E. Luo, S. H. Chan, and T. Q. Nguyen, "Adaptive image denoising by mixture adaptation," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4489–4503, Oct. 2016.

[50] C. Tang, X. Yang, and G. Zhai, "Noise estimation of natural images via statistical analysis and noise injection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 8, pp. 1283–1294, Apr. 2015.

[51] L. Dong, J. Zhou, and Y. Y. Tang, "Noise level estimation for natural images based on scale-invariant kurtosis and piecewise stationarity," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 1017–1030, Feb. 2017.

[52] L. Dong, J. Zhou, and Y. Y. Tang, "Effective and fast estimation for image sensor noise via constrained weighted least squares," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2715–2730, Jun. 2018.

[53] Q. Guo, C. Zhang, Y. Zhang, and H. Liu, "An efficient svd-based method for image denoising," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 5, pp. 868–880, May 2016.

[54] A. C.-C. Yao, "How to generate and exchange secrets," in *Proc. IEEE 27th Annu. Symp. Found. Comput. Sci.*, 1986, pp. 162–167.

[55] Y. Lindell and B. Pinkas, "A proof of security of yao's protocol for two-party computation," *J. Cryptology*, vol. 22, no. 2, pp. 161–188, 2009.

[56] S. Even, O. Goldreich, and A. Lempel, "A randomized protocol for signing contracts," *Commun. ACM*, vol. 28, no. 6, pp. 637–647, 1985.

[57] R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan, "Cryptdb: Protecting confidentiality with encrypted query processing," in *Proc. 23rd ACM Symp. Operating Syst. Principles*, 2011, pp. 85–100.

[58] D. Beaver, "Efficient multiparty protocols using circuit randomization," in *Proc. Annu. Int. Cryptology Conf.*, 1991, pp. 420–432.

[59] M. D. Cock, R. Dowsley, C. Horst, R. Katti, A. Nascimento, W.-S. Poon, and S. Truex, "Efficient and private scoring of decision trees, support vector machines and logistic regression models based on pre-computation," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 2, pp. 3411–3425, Mar./Apr. 2017.

[60] R. Livni, S. Shalev-Shwartz, and O. Shamir, "On the computational efficiency of training neural networks," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst. - Vol. 1*, 2014, pp. 855–863.

[61] R. Canetti, "Security and composition of multiparty cryptographic protocols," *J. Cryptology*, vol. 13, no. 1, pp. 143–202, 2000.

[62] C. Guan, A. Mohaisen, Z. Sun, L. Su, K. Ren, and Y. Yang, "When smart TV meets CRN: Privacy-preserving fine-grained spectrum access," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst.*, 2017, pp. 1105–1115.

[63] J. H. Ziegeldorf, J. Metzke, J. Rüth, M. Henze, and K. Wehrle, "Privacy-preserving HMM forward computation," in *Proc. 7th ACM Conf. Data Appl. Secur. Privacy*, 2017, pp. 83–94.

[64] Y. Wang and J. Morel, "Can a single image denoising neural network handle all levels of gaussian noise?" *IEEE Signal Process. Lett.*, vol. 21, no. 9, pp. 1150–1153, Sep. 2014.

[65] V. Kolesnikov and T. Schneider, "Improved garbled circuit: Free XOR gates and applications," in *Proc. Int. Colloquium Automata Lang. Program.*, 2008, pp. 486–498.

[66] M. Bellare, V. T. Hoang, S. Keelveedhi, and P. Rogaway, "Efficient garbling from a fixed-key blockcipher," in *Proc. IEEE Symp. Secur. Privacy*, 2013, pp. 478–492.

[67] B. Pinkas, T. Schneider, N. P. Smart, and S. C. Williams, "Secure two-party computation is practical," in *Proc. 15th Int. Conf. Theory Appl. Cryptology Inf. Secur.: Adv. Cryptology*, 2009, pp. 250–267.

[68] T. Schneider, *Engineering Secure Two-Party Computation Protocols - Design, Optimization, and Applications of Efficient Secure Function Evaluation*. New, NY, USA: Springer, 2012.

[69] X. Wang, Y. Peng, L. Lu, Z. Lu, M. Bagheri, and R. M. Summers, "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3462–3471.

[70] S. Li, H. Yin, and L. Fang, "Group-sparse representation with dictionary learning for medical image denoising and fusion," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 12, pp. 3450–3459, Dec. 2012.

[71] J. Bai, S. Song, T. Fan, and L. Jiao, "Medical image denoising based on sparse dictionary learning and cluster ensemble," *Soft Comput.*, vol. 22, no. 5, pp. 1467–1473, Mar. 2018.

[72] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising with multi-layer perceptrons, part 1: Comparison with existing algorithms and with bounds," *CoRR*, vol. abs/1211.1544, pp. 1–38, 2012.

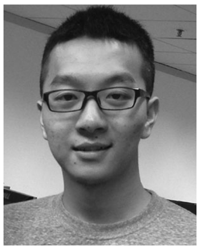
[73] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[74] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

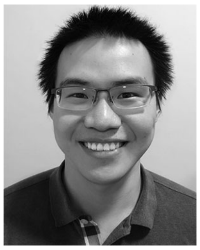
[75] K. Zhang, W. Zuo, and L. Zhang, "Ffdnet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.

Authorized licensed use limited to: POLO BIBLIOTECARIO DI INGEGNERIA. Downloaded on June 02,2021 at 15:51:59 UTC from IEEE Xplore. Restrictions apply.

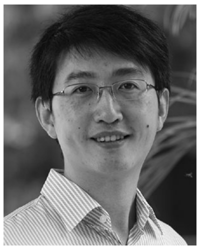
Yifeng Zheng received the BE degree in information engineering from South China University of Technology, Guangzhou, China, in 2013. He is currently working toward the PhD degree in the Department of Computer Science, City University of Hong Kong, Hong Kong, China. From 2014 to 2015, he worked as a research assistant with the Department of Computer Science, City University of Hong Kong, Hong Kong, China. His current research interests include security in cloud media applications, security in IoT applications, and privacy-aware computing. He received the Outstanding Teaching Awards for Teaching Assistants, the Outstanding Academic Performance Award, and the Chow Yei Ching School of Graduate Studies Scholarship from City University of Hong Kong, in 2018.



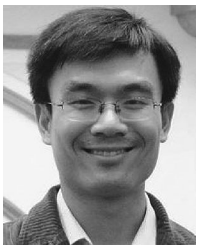
Huayi Duan received the BS degree with first class honors from City University of Hong Kong, Hong Kong, in 2015. He is currently working toward the PhD degree in the Department of Computer Science, City University of Hong Kong, Hong Kong. His research interests include network security and cloud security.



Xiaoting Tang received the BS degree from City University of Hong Kong, in 2016. He is currently working toward the MS degree in the Department of Computer Science, Brown University, United States of America. His research interests include computation security and cloud security.



Cong Wang received the BE degree in electronic information engineering and ME degree in communication and information system, both from Wuhan University, China, and the PhD degree in the electrical and computer engineering from Illinois Institute of Technology. He is currently an associate professor with the Department of Computer Science, City University of Hong Kong. His current research interests include data and computation outsourcing security in the context of cloud computing, blockchain and decentralized application, network security in emerging Internet architecture, multimedia security, and privacy-enhancing technologies in the context of big data and IoT. He is one of the Founding Members of the Young Academy of Sciences of Hong Kong. He received the Outstanding Supervisor Award in 2017 and the President's Awards from City University of Hong Kong in 2016. He is a co-recipient of the Best Student Paper Award of IEEE ICDCS 2017, the Best Paper Award of IEEE ICPADS 2018, MSN 2015 and CHINACOM 2009. His research has been supported by multiple government research fund agencies, including National Natural Science Foundation of China, Hong Kong Research Grants Council, and Hong Kong Innovation and Technology Commission. He serves/has served as associate editor for the *IEEE Transactions on Dependable and Secure Computing*, the *IEEE Internet of Things Journal* and the *IEEE Networking Letters*, and TPC co-chairs for a number of IEEE conferences/workshops. He is a senior member of the IEEE, and member of the ACM.



Jiantao Zhou received the BEng degree from the Department of Electronic Engineering, Dalian University of Technology, in 2002, the MPhil degree from the Department of Radio Engineering, Southeast University, in 2005, and the PhD degree from the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, in 2009. He held various research positions with the University of Illinois at Urbana-Champaign, the Hong Kong University of Science and Technology, and the McMaster University. He is currently an associate professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau. His research interests include multimedia security and forensics, multimedia signal processing, artificial intelligence and big data. He received four granted U.S. patents and two granted Chinese patents. He has co-authored two papers that received the Best Paper Award at the IEEE Pacific-Rim Conference on Multimedia in 2007 and the Best Student Paper Award at the IEEE International Conference on Multimedia and Expo in 2016. He is an associate editor of the *IEEE Transactions on Image Processing*.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.

Authorized licensed use limited to: POLO BIBLIOTECARIO DI INGEGNERIA. Downloaded on June 02,2021 at 15:51:59 UTC from IEEE Xplore. Restrictions apply.