Abstract—In this paper, we present a novel Secured Outsourced Content Based Image Retrieval solution, which allows looking for similar images stored into the cloud in a homomorphically encrypted form. Its originality is fourfold. In a first time, it extracts from a Paillier encrypted image a wavelet based global image signature. In second, this signature is extracted in an encrypted form and gives no clues about the image content. In a third time, its calculation does not require the cloud to communicate with a trusted third party as usually proposed by other existing schemes. More clearly, all computations required in order to look for similar images are conducted by the cloud only with no extra-communications. To make possible such a computation, we propose a new fast way to compare encrypted data, these ones being encrypted by the same public key or not, and using a recursive relationship in-between Paillier random values when computing the different resolution levels of the image wavelet transform. Experiments conducted in two distinct frameworks: medical image retrieval with as purpose diagnosis aid support, and face recognition for user authentication; indicate that the proposed SOCBIR does not change image retrieval performance.

Index Terms—Homomorphic encryption, secure content based image retrieval, processing of encrypted data, cloud computing.

I. INTRODUCTION

Cloud computing has emerged as a successful paradigm allowing individuals and companies to flexibly store and process large amounts of data without a need to purchase and maintain their own networks and computer systems. In healthcare for example, different initiatives aim at sharing medical images and Personal Health Records (PHR) in between health professionals or hospitals with the help of the cloud [1][2][3]. In order, to take advantage of such volume of data and of the medical knowledge it represents, several Content Based Image Retrieval (CBIR) methods have been introduced to support decision making [4]. In a general way, the basic concept of a CBIR system stands in the retrieval of images into a database that are similar to a request image. In medical imaging, knowing the diagnosis associated to these similar images can give clues to the physician before he accesses or analyses the image. Another application that can take advantage of CBIR corresponds to user authentication based on face recognition, where a user is identified through the comparison of his biometric data with those stored and managed in a database into the cloud by a third party [5]. If there is a match, the user can log into his account or obtain the service provided by the third party.

Beyond their innovative character, such cloud applications must take into account data security issues that are increased especially in terms of confidentiality and privacy. Indeed, data outsourcing involves at the same time that users lose the control on their own data [6]. Recent news show clear evidence that outsourced storage is not safe against privacy threats, these ones being external (e.g., hackers [7]) or internal [5][8]. There is thus an interest to develop Secured Outsourced Content Based Image Retrieval (SOCBIR) methods.

Different approaches have been proposed to secure CBIR methods. They can be differentiated depending on if they work on local or global image signature, and on the way they make interacting encryption with image signature computation and comparison. Regarding the way signatures are handled, one first solution consists in outsourcing an image along with its signature calculated a priori [9]. In this scenario, the server (or equivalently the cloud) only conducts signatures’ comparison. Images are encrypted with algorithms such as AES or 3-DES and cannot be manipulated unless they are decrypted. The CBIR process is thus shared between the client and the server. An alternative to this approach proposes to extract signatures or features directly from encrypted images. To do so, most solutions make use of homomorphic encryption the interest of which is that it allows performing operations (e.g. $+$, $\times$) onto encrypted data with the guarantee that the decrypted result equals the one carried out onto unencrypted data. To the best of our knowledge, Erkin et al. [10] proposed the first privacy-preserving biometric face recognition scheme securing the Eigenfaces recognition algorithm. However, this scheme only ensures the confidentiality of the query image due to the fact the server compares the signature extracted from the encrypted image it received to the ones of images stored in its database under an un-encrypted form. In [11], SIFT features are extracted from an image encrypted with the Paillier homomorphic cryptosystem. This method needs 1-round of communication between the server and the user for features’ extraction and much more to compare image signatures. This scheme has also been proved unfeasible in terms of computational complexity with some security weaknesses [12]. The authors of [12] improve this scheme with the
help of fully homomorphic encryption and secure multiparty computation but with as consequence a great increase of the storage, computation and communication complexities. Another improvement suggested in \[13\], stands on the use of order-preserving encryption (OPE \[14\]) and three non-colluding clouds. It should be noticed that both solutions given in \[12\] and \[13\] do not well preserve the performance of the original SIFT features. To overcome this issue, \[15\] and \[16\] propose a secure SIFT feature set based on two independent cloud servers. If the complexity of communication with the user is reduced, it assumes that the two servers do not collude. In \[17\], the interest is given to SURF features, another kind of local image features, and to their extraction from homomorphically encrypted images. As the previous schemes, this solution requires highly interactive communications between the server and the user in order to extract the image signature by executing certain basic operations in the encrypted domain the server cannot conduct alone without accessing to the data in clear (e.g., division, square root). \[18\] proposed to reduce the communication complexity of \[17\] with the help of somewhat homomorphic encryption (SHE) and two non-colluding cloud servers. But using SHE also limits the number of both addition and multiplication operations one can execute repeatedly over cipher-texts (an issue linked to the SHE depth - see \[19\] for more details). \[20\] \[21\] were the first to introduce a global image signature based SOCBIR. Rather than extracting some local characteristics as SIFT and SURF features, these schemes sums up an image to the distribution of its wavelet coefficients in different sub-bands. Such a signature has been shown more optimal in terms of retrieval performance when the image texture plays a major role, which is often the case in medical imaging.

Beyond the fact that these SOCBIR schemes require communications in-between the user and one or two servers, they compute image signatures in an un-encrypted form. Even if the image content is not available, knowing its SIFT features may give big clues about it. These features are indeed characteristic points of the image. For instance, their positions can contribute to the recognition of objects or person in an encrypted image. We give in Fig. 1 an example of such a situation. There is thus a need to extract image signatures that are themselves encrypted. Such an issue was very recently addressed in \[21\] and by next in \[15\] \[16\]. But, these solutions exploit at least two cloud servers when computing signatures with as consequence the increase of already high communication requirements.

In this paper, we propose a novel global image signature based SOCBIR. Working with the Paillier cryptosystem, it first allows extracting from an encrypted image a signature in an encrypted form, giving thus no clues about the image content. It is based on two original secure signature calculation and comparison algorithms that are fully conducted by the server (or equivalently the cloud) that is to say without communications with a third party (i.e., the user or another server). Both algorithms take advantage of a new way to rapidly compare Paillier encrypted data with no communications, even if the data are encrypted with different public keys. In order to make possible the computation of a signature that corresponds to the histograms of different wavelet sub-bands of the image, we show the existence of a recursive relationship in-between the random values of the Paillier cryptosystem at different resolutions levels of the image wavelet transform.

The rest of this paper is organized as follows. In Section \[II\] we come back on the definition of the homomorphic encryption algorithm of Paillier and show how to use it so as to rapidly compare encrypted data. Section \[III\] presents the proposed Secure Outsourced CBIR scheme, detailing how to extract an encrypted signature from an encrypted image with no communication between the server or a third party (e.g., the user or a second service provider). Its security and complexity are discussed in Section \[IV\]. Performance of the proposed solution are given and compared to the original solution in the clear domain in Section \[V\] considering two application frameworks: diagnosis aid support and user authentication based on facial recognition. Section \[VI\] concludes this paper.

II. PROCESSING OF HOMOMORPHICALLY ENCRYPTED DATA WITH THE PAILLIER CRYPTOSYSTEM

This section first presents the Paillier cryptosystem before introducing how to rapidly compare data encrypted or not with the same public key.

A. Paillier cryptosystem

We opted for the asymmetric Paillier cryptosystem because of its additive homomorphic property \[22\]. In this work, we use a fast version of it defined as follows. Let \((g, K_p), K_s\) be the public/private key pair, such as

\[
K_p = pq \quad \text{and} \quad K_s = (p - 1)(q - 1) \quad (1)
\]

Fig. 1: Illustrative examples issued from \[11\] of SIFT features detection in the plain-text domain (a), (c), and in the Cipher-text domain (b), (d).
where \( p \) and \( q \) are two large prime integers. \( \mathbb{Z}_{K_p} = \{0, 1, ..., K_p - 1\} \) and \( \mathbb{Z}_{K_p}^* \) denotes integers that have multiplicative inverses modulo \( K_p \). We select \( g \in \mathbb{Z}_{K_p}^* \) such as

\[
g^{K_p} - 1 \mod K_p^2 \in \mathbb{Z}_{K_p}^* \tag{2}
\]

The Paillier encryption of a plain-text \( m \in \mathbb{Z}_{K_p} \) into the cipher-text \( c \in \mathbb{Z}_{K_p}^2 \) using the public key \( K_p \) is given by

\[
c = E[m, r] = g^{mK_p}r^{K_p^2} \mod K_p^2 \tag{3}
\]

where \( r \in \mathbb{Z}_{K_p}^* \) is a random integer associated to \( m \) making the Paillier cryptosystem probabilistic or semantically secure. More clearly, depending on the value of \( r \), the encryption of the same plain-text message will yield to different cipher-texts even though the public encryption key is the same. As introduced in [23], it is possible to get a fast version of (3) by fixing \( g = 1 + K_p \) without reducing the algorithm security. By doing so, the encryption of \( m \) into \( c \) requires only one modular exponentiation and two modular multiplications

\[
c = E[m, r] = (1 + mK_p)r^{K_p^2} \mod K_p^2 \tag{4}
\]

As we will see in Section II-B this property will be of importance for the fast comparison of Paillier encrypted data.

Based on the assumption \( g = 1 + K_p \), the decryption of \( c \) using the private key \( K_s \) is such as

\[
m = \frac{(c^{K_s} - 1)K_p^{-1} \mod K_p^2}{K_p} \mod K_p \tag{5}
\]

If we consider two plain-texts \( m_1 \) and \( m_2 \), the additive homomorphic property of the Paillier cryptosystem allows linear operations on encrypted data like addition and multiplication, ensuring that

\[
E[m_1, r_1].E[m_2, r_2] = E[m_1 + m_2, r_1r_2] \tag{6}
\]

\[
E[m_1, r_1]^{m_2} = E[m_1, m_2, r_1r_2^{m_2}] \tag{7}
\]

### B. Comparing Paillier encrypted data

Even though the additive property of the Paillier cryptosystem allows linear operations, it cannot conduct more complex operations like data comparison or modulo reduction. These operations are however essential in signal processing.

In our concern and as we will see in Section III comparison of encrypted data is a core operation in SOCBIR. Two distinct classes of solutions can be considered. The first one, which is also the main one studied since a long time, regroups methods that are based on a protocol; a set of interactions between two parties that do not collude. Many secure protocols are known for comparing two integers, the so called millionaires’ problem. The first protocol was proposed by Yao [24] in 1982. It corresponds to the well-known garbled circuit and has been improved many times since. One of the most efficient implementation based on homomorphic encryption is described by T. Veuugen [25]. But, this one, as all the others, requires many communications between parties so as to conduct one comparison. The second kind of approach is communication free and has been recently introduced by Hsu et al. in [11].

The solution we propose is an improvement of this approach. It allows comparing two encrypted data, encrypted or not with the same public key, in a less complex way. It takes into account the fast implementation of the Paillier cryptosystem of Section II-A.

#### • Comparing data encrypted with the same public key

Before presenting our solution, let us come back on the proposal of Hsu et al. [11].

Let us consider a client-server relationship where the server possesses two cipher-texts \( c_1 = E[m_1, r_1] \) and \( c_2 = E[m_2, r_2] \) encrypted by the client; \( m_1 \) and \( m_2 \) are two integer values the server wants to compare knowing only \( c_1 \) and \( c_2 \). The basic idea of [11] is not to compare directly data but to establish their relationship (smaller, greater or equal) through the quantization of data from their encrypted versions. In a first time, the client sends along with \( c_1 \) and \( c_2 \), a set of thresholds \( \{T_i\}_{1 \leq i \leq N} \) that quantize the dynamic range of both integer values \( m_1 \) and \( m_2 \). In practice, the client chooses an increasing sequence of randomly distributed thresholds \( \{T_i \in \mathbb{Z}_{K_p}\}_{1 \leq i \leq N} \) and sends \( E(T_i, r_i) \) \( \{1 \leq i \leq N \} \) to the server. To compare data, the server has first to identify the interval to which belongs each message \( m_u, u \in \{1, 2\} \). Herein, it is considered that \( m_u \) belongs to the \( k_u \)th interval, if the closest threshold of \( m_u \) is \( T_{k_u} \).

To identify \( k_u \), the server starts the following iterative process:

1) **Comparison of \( E[m_u, r_u] \) with one threshold \( E[T_i, r_u] \)**

Based on the Paillier cryptosystem property \( E[m_u, r_u].E[a, b] = E[m_u + a, r_u.b] \) (see [11]), it is possible to compute the distance \( D(T_i, m_u) = T_i - m_u \) by iteratively multiplying \( E[m_u, r_u] \) with \( E[1, 1] \) until the product equals \( E[T_i, r_u] \). The number of iterations \( inc \) gives \( D(T_i, m_u) \) (i.e. \( E[T_i, r_u] = E[m_u + inc, r_u] \); \( D(m_u, T_i) = inc \)

2) **Closest threshold of \( m_u \)**

The closest threshold \( T_{k_u} \) is naturally given by the minimum distance in-between \( E[m_u, r_u] \) and all encrypted thresholds \( \{E[T_i, r_u]\}_{1 \leq i \leq N} \). Based on the fact \( E[m_u, r_u]g^{inc} = E[m_u + inc, r_u] \), this whole process can be summed up for any message \( m_u, u \in \{1, 2\} \), by

\[
\begin{align*}
(d_u, k_u) &= \arg \min_{1 \leq i \leq N}(E[T_i, r_u] - E[m_u, r_u]g^{inc}) \mod K_p^2 \\
\end{align*}
\]

where \( d_u \) is the distance between the message \( m_u \) and its closest threshold of index \( k_u \).

After having computed \( (d_1, k_1) \) and \( (d_2, k_2) \) for \( m_1 \) and \( m_2 \), respectively, it is possible to determine whether \( m_1 \leq m_2 \) or not without decrypting them. This is, due to the fact that the same thresholds \( \{T_i\}_{1 \leq i \leq N} \) are used for \( m_1 \) and \( m_2 \). Notice that knowing the relative distance between \( m_1 \) and \( m_2 \) gives no ideas about their respective values. At the same time, the complexity of this approach depends of the number of threshold.

Compared to [11] our proposal does not need an iterative procedure as well as sending several thresholds. On the contrary the user will just have to send to the server two encrypted versions of one single threshold \( T \), i.e., \( E[T, r_1] \) and \( E[T, r_2] \). \( T \) will be used as "value of reference" into the
It is asked to determine if the user when computing and comparing image signatures; and, ii) extract from an encrypted image an encrypted signature of the image corresponds to the set of its wavelet sub-band histograms up to a given decomposition level. In this example, only one decomposition level is considered. $HH$, $HG$, $GH$ and $GG$ represent the approximation subband and the horizontal, vertical and diagonal detail subbands, respectively.

### III. PROPOSED SECURED OUTSOURCED CONTENT BASED IMAGE RETRIEVAL SYSTEM

In this section, we first provide a global view of a CBIR system outsourced into the cloud before exposing how to conduct each of its functionalities in a secure manner.

#### A. From classic CBIR to secure outsourced CBIR

The basic interactions of an outsourced content based image retrieval system are exposed in Fig. 2 in the case of a diagnosis aid support application. Two entities interact: the user (e.g., the physician’s system) sends a query image to the cloud so as to get an idea about the possible diagnosis. When the cloud receives the request, it extracts or computes from the image a signature it next compares to the ones of the images stored in its database. It then returns to the user the most similar images along with their diagnosis. In this scenario, all computations are conducted in clear by the server. They correspond to the image signature extraction and comparison which are the two basic functionalities of a CBIR system.

In this work, we consider the global image signature based CBIR algorithm proposed in [4] which sums up an image into a signature that corresponds to the concatenation of the histograms of the different wavelet sub-bands of the image (see Fig. 3). The computation of this signature relies on two steps:

- Calculation of the Discrete Wavelet Transform (DWT) of the image up to the $d^{th}$ decomposition level.
- Computation of the histograms of the wavelet sub-bands from the level 0 to the level $d$; and concatenation of the histograms so as to obtain the image signature.

The solution we propose and detail in the sequel has been developed under the following constraints: i) zero interaction in-between the server or any other trusted third parties including the user when computing and comparing image signatures; and, ii) extract from an encrypted image an encrypted signature. As we will see, our proposal takes advantage of the additive homomorphic property of the Paillier cryptosystem and of the communication free comparison solutions we derived.
in Section II-B. More clearly, our system allows retrieving similar images with no needs of communications.

B. Wavelet Image Transform in the Paillier Encrypted Domain

The 2D wavelet transform of an image $I$ in the clear domain, considering the wavelet is separable, is defined as 

$$a^d(x, y) = \sum_{l, l' \in \mathbb{Z}} h(2x - y)h(2y - l')a^{d-1}(l, l')$$

$$b^d_u(x, y) = \sum_{l, l' \in \mathbb{Z}} u(2x - y)u'(2y - l')a^{d-1}(l, l')$$

where $a^d(x, y)$ and $b^d_u(x, y)$ are the approximation and detail coefficients of the $d^{th}$ decomposition level, respectively; $(x, y)$ gives the position of the coefficient in the sub-band; $u, u' \in \{h, g\}$ where $h$ and $g$ are the low-pass and high-pass wavelet decomposition filters, respectively; $u \in \{bh, gh, gg\}$ is the index of the detail coefficient sub-bands (i.e., horizontal, vertical and diagonal). In the case $d = 0$, $a^0(x, y)$ corresponds to the image pixel $I(x, y)$.

Due to the fact that the Paillier cryptosystem works with plain-text and cipher-text constituted of positive integers in $\mathbb{Z}_{K_p}$ (where $K_p$ is the cryptosystem public key), all image pixels and image processing algorithm’s parameters should also be represented by integers. Even if we assume that all pixels are integers, i.e., $I(x, y) \in \mathbb{Z}_{K_p}$, their processing may lead to negative values. In order to represent such values in $\mathbb{Z}_{K_p}$, integer values greater than $K_p/2$ will correspond to negative values and the others to positive values.

Similarly, in order to compute DWT in the encrypted domain, the coefficients of the decomposition filters $h$ and $g$, which are real numbers, should be turned into integer values. This can be achieved by scaling and quantizing the coefficients of $h$ and $g$ as follow 

$$G = \lfloor Qg \rfloor; H = \lfloor Qh \rfloor$$

where $\lfloor \rfloor$ is the rounding function and $Q$ is a scaling factor. $Q$ is a parameter defined by the user accordingly to the wavelet transform he wants to exploit.

From this standpoint, the encrypted versions of the wavelet approximation and detail coefficients in the encrypted domain are given as (using (6) and (7))

$$E[A^d(x, y), r^d(x, y)] = \prod_{l, l' \in \mathbb{Z}} E[A^{d-1}(l, l'), r^{d-1}(l, l')]H(2x - y)H(2y - l')$$

$$E[B^d_u(x, y), r^d_u(x, y)] = \prod_{l, l' \in \mathbb{Z}} E[A^{d-1}(l, l'), r^{d-1}(l, l')]W(2x - y)W'(2y - l')$$

where $A^d(x, y)$ and $B^d_u(x, y)$ are the wavelet approximation and detail coefficients of the $d^{th}$ decomposition level, respectively; $W, W' \in \{H, G\}$ where $H$ and $G$ are the expanded version of the low-pass and high-pass wavelet decomposition filters $h$ and $g$, respectively; $u \in \{HG, GH, GG\}$ indicates the index of the detail coefficient sub-bands (i.e., horizontal, vertical and diagonal). Notice that all multiplications and exponentiation are carried out into $\mathbb{Z}_{K_p^2}$.

![Fig. 4: Histogram $H_{C'_u}$ of the coefficient sub-band $C'_u$ in the clear domain, in the case of $K = 5$ classes with a coefficient dynamic such as $C'_u(x, y) \in [C_{min}, C_{max}]$.](image)

\{T_u\}_{0 \leq u \leq 4}$ correspond to the class centers.

C. Calculation of the wavelet sub-band histograms in the encrypted domain

As stated in Section III-A once the image DWT computed, the next step consists in building the histograms of its different wavelet sub-bands. Let us first recall how to calculate a histogram in the clear domain before explaining how to make the same task in the Paillier encrypted domain so as to compute a clear histogram and by next an encrypted histogram.

1) Histogram computation in the clear domain: Let us consider $C'_u(x, y)$ is a wavelet coefficient (either a detail or approximation coefficient) at the position $(x, y)$ in the sub-band $u, u \in \{HH, GH, HG, GG\}$, at the decomposition level $d$. In order to build the histogram $H_{C'_u}$ of the sub-band $C'_u$, the coefficient dynamic is first divided into $K$ uniform intervals or classes of width $\Delta$ (see Fig. 4). By definition, the value $H_{C'_u}(k)$ indicates the number of coefficients $C'_u(x, y)$ the values of which belong to the $k^{th}$ interval of $H_{C'_u}$. More clearly, $H_{C'_u}(k)$ gives the cardinality of the class $C_k$. If $T_k$ denotes the center of the $k^{th}$ interval of $H_{C'_u}$, then the class $C_k$ of $C'_u(x, y)$ is given by:

$$k = \arg \min_p |C'_u(x, y) - T_p|$$

Another way to construct a histogram is to use a single threshold instead of the $K$ as explained before. The idea behind this approach is that the server uses this threshold as a reference point in order to find the membership classes regardless of the quantification step. To do this, we will consider $T$ as a reference point, then the server computes the distance $d_{x,y}$ between $T$ and each wavelet coefficient $C'_u(x, y)$, this distance is given as:

$$d_{x,y} = T - C'_u(x, y)$$

After the computation of the distances, the objective now is to construct the histogram from the distances between the reference points and all the wavelet coefficients. To do this, we suppose that $d_{x,y}$ and $d_{x',y'}$ present the distances of $C'_u(x, y)$ and $C'_u(x', y')$ with $T$ respectively, then it is sufficient to compare $|d_{x,y}|$ and $|d_{x',y'}|$, where $\Delta$ is the quantization step and $\|$ is the floor operator. By doing so, if $|d_{x,y}|/\Delta$ and $|d_{x',y'}|/\Delta$ are equal then $C'_u(x, y)$ and $C'_u(x', y')$ belong to the same class $C_k$ otherwise no. After the comparison between the
distances, the server can build an histogram with $K$ uniform classes, where $K$ is given as $K = |C^d_u|/\Delta$ and $|C^d_u|$ presents the cardinal of the dynamic bounds $C^d_u$.

2) Clear histogram computation in the encrypted domain:
The construction of the above histogram in the encrypted domain imposes to determine the class $C^e_k$ to which belongs a wavelet coefficient $C^e_k(x, y)$ from its encrypted version $C^e_k(x, y) = E[C^d_k(x, y), r^d_k(x, y)]$ where $r^d_k(x, y)$ is a random integer number (see (3)). In other words, we need to compute (16) in the encrypted domain.

To do so, the user has to send along with the encrypted image the encrypted versions of the histogram class center, i.e., $E[T, r^d_k(x, y)]$. Based on this information and using the encrypted data comparison solution (9) of Section II-B, it is possible to determine the interval or class of a the coefficient $C^d_k(x, y)$ from $C^e_k(x, y)$ as follow

$$d_{x,y} = D(C^e_k(x, y), E[T, r^d_k(x, y)]) \tag{17}$$

It is important to underline that in order the cloud server computes (17) based on (9), the user needs to encrypt the threshold $T$ with the appropriate random value $r^d_k(x, y)$ that is to say the same random value associated to the encrypted coefficient $C^e_k(x, y)$. It is however possible to find a recursive relation in-between the values $r^d_k(x, y)$ from the level 0 to the level $d$ by using the Paillier cipher properties (6) and (7).

Basically, $r^d_k(x, y), u \in \{HH, GH_d, GH_d, HG_d\}$, is such as

$$r^d_k(x, y) = \prod_{l,l' \in \mathbb{Z}} r^d_u(l, l')^{W(2x-l)W'(2x-l')} \tag{18}$$

As it can be seen $r^d_k(x, y)$ is independent of the image content. For $d = 0$, $r^d_u(x, y) = r(x, y)$, i.e., the random integer value used to encrypt the image pixel $I(x, y)$.

Based on (17) and (18), the server can build the histogram of one wavelet sub-band ($H_{C^d}$). In order to compare two histograms, $H_{C^d(1)}$ and $H_{C^d(2)}$ of two images $I^{(1)}$ and $I^{(2)}$, respectively, one just has for instance to apply the $L_1$-distance

$$L_1(H_{C^d(1)}, H_{C^d(2)}) = \sum_{1 \leq k \leq K} |H_{C^d(1)}(k) - H_{C^d(2)}(k)| \tag{19}$$

Even though this solution makes use of encrypted histogram class center $E[T, r]$, it suffers from two security issues. The first one stands on the fact the class cardinalities $|H_{C^d(1)}(k)|$ are not encrypted. Along with the awareness the class cardinalities of the histogram are ordered, the histogram $H_{C^d}$ appears thus in a clear form. Even if the coefficient dynamic bounds are unknown, if one has some a priori knowledge about the signal distribution, he or she will have big clues about the content of the encrypted data. For instance, it is well known that detail wavelet coefficients follow a Gaussian or Laplacian distribution centered onto zero. We give in Fig. 5 a reconstruction attack example where the pirate replaced the encrypted detail coefficients by approximated values depending on the class they belong to. As it can be seen, even though the approximation sub-band is unknown, it is possible to identify that the encrypted image corresponds to the face of a person. There is thus a need to secure the histogram itself. The second weakness is in relation with the comparison algorithm depicted in Section II-B. If the pirate has an idea about the dynamic bounds of $C^d_u$, and if these bounds are part of the encrypted class centers, then he will be able to determine the real value of these centers. We discuss how to address these issues in the next section.

D. Encrypted wavelet coefficient histogram

The above solution allows constructing a $K$ class histogram $H_{C^d}$ in a clear form from an encrypted wavelet sub-band $C^d_u$. The approach we propose now aims at building a secure histogram $H^E_{C^d}$ the class cardinalities of which are in an encrypted form (i.e. $E[H_{C^d}(k), r_k]_{1 \leq k \leq K}$) and the class centers of which cannot be obtained. It is based on two main steps. The first one consists in building a noisy histogram $H^N_{C^d}$ from the encrypted image wavelet coefficients; histogram from which the secure histogram $H^E_{C^d}$ will be computed, in a second time.

The noisy histogram construction has as purpose to secretly break the distribution of the wavelet coefficients onto a larger dynamic, while giving access to an histogram the class cardinalities of which are in a clear form (i.e., un-encrypted). Our solution is equivalent to the addition of a noise to the wavelet sub-band coefficients and to compute the histogram of the resulting random variable as above. Let us consider one wavelet coefficient $C^d_u(x, y)$ and $N(x, y)$ a uniformly distributed integer in the range $[N_{min}, N_{max}]$ such as $N_{max} > C_{max}$ where $C_{max}$ is the absolute maximum value a wavelet coefficient can take in a subband. The addition of $N(x, y)$ to $C^d_u(x, y)$ leads to a random variable $C^d_u(x, y) + N(x, y)$, the distribution of which is quite closed to a uniformly distributed random variable. In Fig. 6 we illustrate the result of such a procedure in the case of a centered Gaussian random variable $X$ of standard deviation $\sigma = 10$ with a uniform random variable $N$ in the range $[0, 0]$.

The density probability $f_Y$ of the resulting random variable $Y = X + N$ takes the form

$$f_Y(t) = \frac{1}{(b-a)2\pi\sigma^2} \int_a^b \exp\left(-\frac{(x-t)^2}{2\sigma^2}\right) dx \tag{20}$$

From this stand point it is possible to compute the histogram of the coefficients $C^d_u(x, y)$ in the encrypted domain as in the previous subsection. To do so and assuming that the dynamic
of $C_{d,u}^{N}$ is subdivided into $K'$ intervals, the user sends along with his encrypted image a set of $K'$ encrypted class centers, one set per wavelet coefficient (see Section III-C2). In our approach and in order not modifying the encrypted image, which could be used for other purposes than CBIR, it is asked the user to add $N(x,y)$ to the histogram class centers instead of the wavelet coefficient. Adding the noise to the coefficients or to the class centers is equivalent. More clearly, for one coefficient $C_{d,u}^{a}(x,y)$, the user send the set of encrypted class centers $\{E[T_k + N(x,y), r_k^{u}(x,y)]\}_{1 \leq k \leq K'}$. $H_{C_{d,u}^{N}}$ will be constructed by the server using these sets of encrypted class centers. Notice, that because $H_{C_{d,u}^{N}}$ is a uniform histogram, it gives no clues to the server about the histogram of $C_{d,u}^{a}$ as illustrated in the example given in Fig. 6. The next step is how to derive the secure histogram $H_{C_{d,u}^{N}}$ of which class of which is $K$, from this noisy histogram of $K'$ classes.

The strategy we propose in order to derive $H_{C_{d,u}^{N}}$ from $H_{C_{d,u}^{N}}$ is similar to Private Information Retrieval (PIR) [23]. It works as follows. Let us consider the computation of the cardinality of the $k^{th}$ class of the encrypted histogram $H_{C_{d,u}^{N}}^{E}(k)$. In a first time and for one coefficient $C_{d,u}^{a}(x,y)$, the user generates a vector $P_{x,y}^{T_k}$ of $K'$ components; components that correspond to the encryption of 0 with different random values (i.e., $E[0, r_k]$) except its $(k + N(x,y))^{th}$ component which is fixed to $E[1, r_u]$. In fact, $P_{x,y}^{T_k}$ indicates for the corresponding coefficient the correct position of the $k^{th}$ class accordingly to the $K'$ classes of $H_{C_{d,u}^{N}}^{E}(k)$ (Fig. 7 illustrates such a mapping between the histogram classes of $H_{C_{d,u}^{N}}^{E}(k)$ and $H_{C_{d,u}^{N}}(k)$). On its side, the server generates a vector $S_{x,y}$ the $K'$ components of which are set to 0 except the $l^{th}$ one which indicates the class of $H_{C_{d,u}^{N}}^{E}(k)$ to which belongs the coefficient $C_{d,u}^{a}(x,y)$. Then, it computes the inner product between $S_{x,y}$ and $P_{x,y}^{T_k}$:

$$S_{x,y}.P_{x,y}^{T_k} = \begin{cases} E[1, r_u] & \text{if } C_{d,u}^{a}(x,y) \in C_k \\ E[0, r_u] & \text{otherwise} \end{cases}$$

(21)

we recall that $C_k$ is the $k^{th}$ class of the histogram of $C_{d,u}^{a}$. More clearly, if $C_{d,u}^{a}(x,y)$ belongs to the $k^{th}$ class of the histogram then $S_{x,y}.P_{x,y}^{T_k}$ equals to $E[1, r_u]$. On the contrary, it equals $E[0, r_u]$. By doing so, we remove the noise added to the class centers without revealing if $C_{d,u}^{a}(x,y)$ belongs the $k^{th}$ class of $C_{d,u}^{N}$. Finally, in order to compute $H_{C_{d,u}^{N}}^{E}(k)$, i.e., the encrypted cardinality of $C_k$, the server just has to multiply the results of the inner products obtained for a sub-band taking advantage of the homomorphic properties of the Paillier cryptosystem (see [7]) that is to say

$$H_{C_{d,u}^{N}}^{E}(k) = \prod_{u} S_{x,y}.P_{x,y}^{T_k} = \prod_{u} E[q_{x,y}, r_u] = E[\sum_{u} q_{x,y}, \prod_{u} r_u]$$

(22)

where $q_{x,y} \in \{0,1\}$.

E. Comparison of two encrypted wavelet coefficient histograms

As exposed in Section III-C2, the comparison of two images $I^{(1)}$ and $I^{(2)}$ stored into the cloud in their encrypted form by two users $U_1$ and $U_2$, respectively, stands on the computation of the $L^1$-distance between the image wavelet sub-band histograms (see [19]).

Let us thus consider two encrypted histograms $H_{C_{d,u}^{E}(1)}$ and $H_{C_{d,u}^{E}(2)}$ extracted from the encrypted version of $I^{(1)}$ and $I^{(2)}$, respectively. At this stand point, it is important to notice that $H_{C_{d,u}^{E}(1)}$ and $H_{C_{d,u}^{E}(2)}$ correspond to the cardinalities $H_{C_{d,u}^{E}(1)}$ and $H_{C_{d,u}^{E}(2)}$ encrypted with different random values and different public keys. It is however possible to compute $D^{N}(H_{C_{d,u}^{E}(1)}, H_{C_{d,u}^{E}(2)}) = D(H_{C_{d,u}^{E}(1)}, H_{C_{d,u}^{E}(2)})$ using the second strategy we proposed in Section II-B.

In order to measure the cardinality difference between two classes, i.e., $H_{C_{d,u}^{E}(1)}(k)$ and $H_{C_{d,u}^{E}(2)}(k)$, this solution requires the comparison of these two quantities with a reference value $p_k$; value $U_1$ and $U_2$ a priori agreed on; that should be encrypted with the same random value as in $H_{C_{d,u}^{E}(1)}(k) = E[H_{C_{d,u}^{E}(1)}(k), r_k]$ and $H_{C_{d,u}^{E}(2)}(k) = E[H_{C_{d,u}^{E}(2)}(k), r_k]$. The problem here is that $r_1^1$ and $r_2^2$ result from multiple operations involved in the computation of $H_{C_{d,u}^{E}(1)}(k)$ and $H_{C_{d,u}^{E}(2)}(k)$, respectively (see Section III-D). In order to make this step possible without introducing interactions between the server and the users, these ones have to send some further pieces of information they ask the server to process along with the computation of the secure histogram procedure so as to get $E[p_k, r_k^1]$ and $E[p_k, r_k^2]$. To do so, for one coefficient $C_{d,u}^{a}(x,y)$, both users generate a vector $P_{z,y}$ the $K'$ component of which correspond to the value $p_k$ encrypted with the same random values as for $P_{x,y}^{T_k}$ (i.e., the vector that allows the mapping in between the classes of the noisy histogram and the clear
histogram - see Section II-B). During the computation of $H_{C^d_u(k)}^{E}(1)$, for instance, the server is asked for each coefficient to compute along with $P_{x,y}$ the inner product

$$S_{x,y}P_{x,y} = \left\{ \begin{array}{ll} E[p_k, r_u] & \text{if } C^d_u(x, y) \in C_k \\ E[p_k, r_z] & \text{otherwise} \end{array} \right. \quad (23)$$

Then and as for computing $H_{C^d_u(k)}^{E}(k)$, it has to multiply the inner products’ results in order to get

$$E[\sum p_k, r_k^1] = \prod S_{x,y}P_{x,y} = \prod E[p_k, r_u] = E[\sum p_k, \prod r_u] \quad (24)$$

Due to the fact the same random values have been used either for computing $H_{C^d_u(k)}^{E}(1)$ and $E[\sum p_k, r_k^1]$, we have $r_k^1 = \prod r_u$. Following the same procedure with the image of the second user, the server will get access to $E[\sum p_k, r_k^2]$. Assuming the images are of same dimensions, the $L^2$-distance in-between two classes is

$$D^2(H_{C^d_u(k)}^{E}(1), H_{C^d_u(k)}^{E}(2)) = D(H_{C^d_u(k)}^{E}(1), \sum p_k) - D(H_{C^d_u(k)}^{E}(2), \sum p_k) = H_{C^d_u(k)}^{E}(1) - H_{C^d_u(k)}^{E}(2) \quad (25)$$

and the $L^1$ distance between the two encrypted histograms

$$D^1(H_{C^d_u(k)}^{E}(1), H_{C^d_u(k)}^{E}(2)) = \sum D^1(H_{C^d_u(k)}^{E}(1), H_{C^d_u(k)}^{E}(2)) \quad (26)$$

F. Summary of the procedure

To sum-up, our Secured Outsourced Content Based Image Retrieval requires that one user sends with his encrypted image the following different pieces of information:

- In order the server computes the noisy histograms $H_{C^d_u}$ for each DWT sub-bands, the user needs to provide for each coefficient, $K'$ Paillier encrypted class-centers.
- In order the server calculates the encrypted cardinality of the $k^{th}$ class of the encrypted histogram $H_{C^d_u(k)}^{E}(k)$, and to compare it with the one of another image, the user has to upload for each coefficient
  - One vector $P_{x,y}$ of $K'$ components that maps the $k^{th}$ class of the histogram in the clear domain with the classes of the noisy histogram; $P_{x,y}$ will be exploited by the server so as to calculate $H_{C^d_u(k)}^{E}(k)$
  - One vector $P_{x,y}$ that contains a reference value the different users agreed on and which is encrypted $K'$ times with the same random values as for $P_{x,y}$, vector the server will use in order to compute the $L^1$ distance in between encrypted histograms of different images.

Once all data have been received, the server can compute in a differered time the different encrypted histograms with no-needs to interact with a trusted third party or the user.

To conclude, in the case of a dyadic separable wavelet transform and images of even dimensions $n.m$, as well as a wavelet transform up to $d$ decomposition levels while considering that a sub-band histogram in the clear domain (i.e., $H_{C^d_u}$) is constituted of $K$ classes and that the noisy histogram stands on $K'$ classes, the number of Paillier encrypted integer values that the user has to sends along with the image is

$$m.n.K' + 2.K'.K[3 \sum_{i=1}^{d} \frac{m}{2^i} \frac{n}{2^i}] = m.n.K'(2K + 1) \quad (27)$$

IV. DISCUSSION

A. Scheme security

The security of our SCOBIR approach relies on the construction of secure wavelet sub-band histograms, the cardinalities of which are encrypted (i.e., $H_{C^d_u}^{E}(k) = \{E[H_{C^d_u(k)}^{N}(k)]\}_{1 \leq k \leq K}$), from a noisy histogram. Such a noisy histogram $H_{C^d_u}^{N}$ is achieved by adding a uniformly distributed noise to the histogram class centers; noise the dynamic of which is much larger than the one of wavelet coefficients (see Section III-D). Even if $H_{C^d_u}^{N}$ is built by the server, i.e., this one knows to which class of $H_{C^d_u}^{N}$ belongs one encrypted coefficient, it has no means to identify the real class of $C^d_u$. Indeed, it does not know the class center values of $H_{C^d_u}^{N}$, which are encrypted, and has no idea of the mapping in between the $K'$ histogram classes of $H_{C^d_u}^{N}$ with the $K$ ones of the clear histogram (i.e., $H_{C^d_u}$). The probability the server guess the correct class of a coefficient $C^d_u(x, y)$ is $\frac{1}{K'}$. In fact, adding the noise to the class centers just shifts the $K$ classes of $H_{C^d_u}$ over the $K'$ ones of $H_{C^d_u}^{N}$. Knowing the class of $C^d_{ue}$ in $H_{C^d_u}^{N}$ only informs the server that $H_{C^d_u}^{N}$ has been shifted around this position for this coefficient, but this one is not able to exactly identify the class of $C^d_u(x, y)$. We give in Fig. 8 the result of the reconstruction attack depicted in Section III-C2 where a pirate replaces encrypted coefficients that are in the same histogram class by the same arbitrary value. As shown, the image remains unintelligible. Even if the server has some a priori knowledge about the distribution of the wavelet coefficient sub-bands, it cannot deduce any information from the noisy histogram.

Regarding the secure wavelet sub-band histogram $H_{C^d_u}^{E}(k)$, due to the fact the cardinalities of its classes are Paillier encrypted, the server has no clues about the class center ordering. Notice also that, based on the way $H_{C^d_u}^{E}(k)$ is derived from $H_{C^d_u}^{N}$ (see Section III-D), the server is not able to identify the $H_{C^d_u}^{E}(k)$ class of an encrypted coefficient $C^d_u(x, y)$.

B. Complexity and memory needs

As stated in Section III-F in the case of a dyadic wavelet transform and an image of even dimensions $n.m$, the user has to encrypt $n.m + n.m.K'(2K + 1)$ integer values (the image and the ancillary data) and send them to the server. These data will be stored in the server database. The associated computation complexity is bounded by $O(m.n.K'(2K + 1))$. This complexity is important but data are sent once. Notice also that all these pieces of information can be used for other purposes than SCOBIR or for computing SOCBIR with different wavelet transforms, for example.

On the server side, the computation complexity stands on the wavelet transform of the images in the encrypted domain,
the construction of the different histograms and on the \( L^1 \)-distance computation between signatures. The computation complexity of the two latter corresponds in fact to the number of comparisons the server has to do which is bounded by \( O(m.n.log_2(K')) \) along with the number of modular multiplications between the results of the histogram inner-products that is bounded by \( O(m.n.K) \). The server need also to perform \( O(m.n.K) \) multiplications to compute the cardinality of one class of the encrypted histogram. To compute the \( L^1 \)-distance in-between signatures, \( O(LK) \) substractions are required, where \( L \) is the size of database. Regarding the computation of the wavelet transform, this one imposes to conduct \( m.n.log_2(K') + 2.m.n.K \) multiplications and additions. As a consequence the computation complexity at the server side is bounded by \( O(m.n.log_2(K') + 2.m.n.K) \) modular multiplications.

Compared to the scheme presented in [21] which allows extracting a secure global signature from a Paillier encrypted image but with the help of two cloud providers, the complexity of which in terms of computation is of \( O(m.n) \) encryption for the user and \( O(m.n.K + L.K + L) \) encryptions for both servers, our approach is of \( K'(2K + 1) \) times greater in terms of computation complexity for the user and \( log_2(K') + K \) times greater for the server. However our proposal uses only one server and is free of communication rounds while [21] require 3 communication rounds (i.e., data transmission, signature computation and signature comparison).

V. EXPERIMENTAL RESULTS

A. Test databases and performance criteria

Our SOCBIR scheme has been experimented considering two application frameworks: diagnosis aid support and authentication based on face recognition (see Section III-B). In both cases, data are stored encrypted in the cloud. For a new image (a medical image to interpret or a user to recognize), the user system encrypts it and sends it to the server along with all pieces of information depicted in Section III-B. The server will send back the most similar images it will have identified in its database.

We give in Fig. 9 some image samples issued from our two test image databases. The medical database regroups 1200 8 bit depth retina images of \( 2240 \times 1488 \) pixels associated with pieces of information about the lesions present into each image. The face recognition database is issued from the Olivetti Research Laboratory of Cambridge, UK. It contains 400 images of \( 92 \times 122 \) pixels that correspond to 40 distinct subjects.

In both use-cases, the performance of our SOCBIR scheme is evaluated in terms of CBIR mean precision which corresponds to the rate of returned images with the same pathology or with the same person as in the query image. As we will see, this measure will vary upon the number of classes considered into the histograms.

B. SOCBIR scheme performance with medical images

In this experiment, the 2D Haar wavelet transform (HWT) has been considered. In order to make its computation possible in the encrypted domain up to the 2\(^{nd}\) resolution level (i.e. \( d = 0, 1, 2 \)), its decomposition filters’ coefficients have been turned into integer values by fixing the quantization factor \( Q \) of \( [12] \) to 4 (see Section III-B). The signature of one image corresponds to the sub-band histograms up to a given resolution level, including the histogram of the approximation images. The face recognition database is issued from the Olivetti Research Laboratory of Cambridge, UK. It contains 400 images of \( 92 \times 122 \) pixels that correspond to 10 photographs of 40 distinct subjects.

In both use-cases, the performance of our SOCBIR scheme is evaluated in terms of CBIR mean precision which corresponds to the rate of returned images with the same pathology or with the same person as in the query image. As we will see, this measure will vary upon the number of classes considered into the histograms.
be seen our SOCBIR scheme has the same performance than the 800 other images constitute the server database. As it can images of our retina image test set as user request images;

200 quantization step values (i.e., $\Delta$) send. In the case images are Paillier encrypted with large prime decomposition levels (i.e., $d$) approach in the clear domain considering different wavelet image, the server returns the five most similar images it finds in for a mean precision at five, that is to say that, for one request histograms are uniformly distributed as expected.

Before detailing the retrieval performance of our scheme, let us first come back on the amount of data the user has to send. In the case images are Paillier encrypted with large prime numbers $p$ and $q$ such that their product, i.e., the public key $K_p$ (see (1)), stands on 1024 bits, then one encrypted integer will be 2048 bit encoded. A direct consequence of such memory needs is that it is difficult to conduct large experimentations with such public key size on a single computer. In order to reduce these needs, all the following tests have been conducted with a public key of 16 bits. The retrieval performance of our scheme with a key of 1024 bits can be directly derived from these results as they are independent of the public key size. Results will be the same. We give in Fig. 10, a view of an encrypted retina image accompanied of the noisy and encrypted histograms of its vertical detail sub-band of its first wavelet decomposition level. It can be seen that these histograms are uniformly distributed as expected.

Fig. 11 provides the retrieval performance of our scheme for a mean precision at five, that is to say that, for one request image, the server returns the five most similar images it finds in its database. We also compare our scheme with the same CBIR approach in the clear domain considering different wavelet decomposition levels (i.e., $d = \{0, 1, 2\}$) and several histogram quantization step values (i.e., $\Delta \in \{1, 2, 4, 8, 16, 32, 64\}$). Notice also that all curves are plotted in average, using 200 images of our retina image test set as user request images; the 800 other images constitute the server database. As it can be seen our SOCBIR scheme has the same performance than the CBIR scheme in the clear domain. Such a result can be explained based on the fact the filter coefficients of the Haar wavelet transform in the encrypted and the clear domains are equivalent. More clearly, the coefficient expansion (12) does not impact the wavelet transform precision. This may not be the case with wavelet transforms the filter coefficients of which are real numbers. Indeed, a loss of precision in the wavelet coefficient computation could reduce the performance retrieval. Beyond, if in the clear domain one may work with $\Delta = 1$, i.e., working with the most accurate histogram based signature, using values of $\Delta$ at least 16 times greater do not impact so much retrieval performance. Let us also recall that the smallest $\Delta$, the more precise is the histogram and the greater is the number of encrypted histogram class centers the user has to send along with his image.

C. SCOBIR performance for face recognition

This experiment has been conducted with the same parametrization as the previous one. Again, our test image database has been split into two sets: 200 facial images are used as request image set while the 200 others are used as reference images by the server. The mean precision at five of our scheme and the equivalent approach in the clear domain is given in Fig. 12 in average, depending on different quantization step values (i.e., $\Delta \in \{1, 2, 4, 8, 16, 32, 64\}$ and decomposition levels (i.e., $d = \{0, 1, 2\}$). The response of our scheme to one request image is also illustrated in Fig. 13. Again, one can see that our SCOBIR approach achieves the same performance as the CBIR approach in the clear domain whatever the quantization step and the level of decomposition. Based on the fact facial images are much smaller than retina images, the amount of information the user has to send to the server is reduced. This is the same for the time computation when comparing images.

VI. Conclusion

In this paper, we have proposed a novel Secure Outsourced CBIR method, which allows completely carrying out a search in an encrypted image database maintained by a server. Contrarily to actual homomorphic encryption based CBIR schemes, it allows the extraction of an encrypted global image signature and does not require extra communications between the user and the server or a trusted third party during the search computation. It exploits a secured histogram procedure we proposed so as to build encrypted wavelet coefficient histograms from an encrypted image taking advantage of a fast comparison of Paillier encrypted data and a noisy histogram. Experimental results show that our SCOBIR scheme achieves the same performance as the equivalent CBIR approach in the clear domain. If our solution guarantees data confidentiality and good retrieval performance with zero communication, its use in real practice remains limited due to the fact it requires the user to send Gigabits of encrypted data when using a public key of 1024 bits so as to be secure. Future works will focus on reducing this amount of information.
This work has been supported by the French national research agency through the LabCom SePEMeD, ANR-13-LAB2-0006-01, and by the Brittany Region Council, ARED AR-MEM.

ACKNOWLEDGMENT

REFERENCES

[1] R. MARCU, D. POPESCU, and I. DANIœLÁ, “Healthcare integration based on cloud computing,” UPB Scientific Bulletin, Series C, vol. 77, 2015.

[2] L. A. B. Silva, “Sharing medical imaging over the cloud services,” International Journal of Computer Assisted Radiology and Surgery, vol. 8, no. 3, pp. 323–333, May 2013.

[3] J. Vincent, W. Pan, and G. Coatrieux, “Privacy protection and security in e-health cloud platform for medical image sharing,” in Advanced Technologies for Signal and Image Processing (ATSIP), 2016 2nd International Conference on. IEEE, 2016, pp. 93–96.

[4] G. Quellec, M. Lamard, G. Cazuguel, B. Cochener, and C. Roux, “Wavelet optimization for content-based image retrieval in medical databases,” in Medical image analysis, vol. 14, no. 2. Elsevier, 2010, pp. 227–241.

[5] J. A. Halderman, S. D. Schoen, N. Heninger, W. Clarkson, W. Paul, J. A. Calandrino, A. J. Feldman, J. Appelbaum, and E. W. Felten, “Lest we remember: cold-boot attacks on encryption keys,” Communications of the ACM, vol. 52, no. 5, pp. 91–98, 2009.

[6] R. Chow, P. Golle, M. Jakobsen, E. Shi, J. Staddon, R. Musauka, and J. Molina, “Controlling data in the cloud: outsourcing computation without outsourcing control.” ACM, 2009, pp. 85–90.

[7] (2014) National Vulnerability Database. cve statistics. [Online]. Available: http://web.nvd.nist.gov/view/vuln/statistics

[8] D. Lewis, “icloud data breach: Hacking and celebrity photos,” Forbes Online, 2014.

[9] W. Lu, A. Swaminathan, A. L. Varna, and M. Wu, “Enabling search over encrypted multimedia databases,” IS&T/SPIE Electronic Imaging, pp. 725418–725418, 2009.

[10] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeiser, I. Lagendijk, and T. Toft, “Privacy-preserving face recognition,” International Symposium on Privacy Enhancing Technologies Symposium, pp. 235–253, 2009.

[11] T.-Y. Hsu, C.-S. Lu, and S.-C. Pei, “Image feature extraction in encrypted domain with privacy-preserving sift,” IEEE Transactions on Image Processing, vol. 21, no. 11, pp. 4593–4607, 2012.

[12] M. Schneider and T. Schneider, “Notes on non-interactive secure comparison in image feature extraction in the encrypted domain with privacy-preserving sift,” Proceedings of the 2nd ACM workshop on Information hiding and multimedia security, pp. 135–140, 2014.

[13] Z. Qin, J. Yan, K. Ren, C. W. Chen, and C. Wang, “Towards efficient privacy-preserving image feature extraction in cloud computing,” in Proceedings of the 22nd ACM international conference on Multimedia, ACM, 2014, pp. 497–506.

[14] A. Boldyreva, N. Chenette, Y. Lee, and A. Oneill, “Order-preserving symmetric encryption,” in Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 2009, pp. 224–241.

[15] Z. Qin, J. Yan, K. Ren, C. W. Chen, and C. Wang, “Secsift: Secure image sift feature extraction in cloud computing,” ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol. 12, no. 4s, p. 65, 2016.

[16] Q. Wang, S. Hu, K. Ren, J. Wang, Z. Wang, and M. Du, “Catch me in the dark: Effective privacy-preserving outsourcing of feature extractions over image data,” in Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on. IEEE, 2016, pp. 1–9.

[17] Y. Bai, L. Zhuo, B. Cheng, and Y. F. Peng, “Surf feature extraction in encrypted domain,” in 2014 IEEE International Conference on Multimedia and Expo (ICME). IEEE, 2014, pp. 1–6.

[18] Q. Wang, S. Hu, J. Wang, and K. Ren, “Secure surfing: Privacy-preserving speeded-up robust feature extractor,” in Distributed Computing Systems (ICDCS), 2016 IEEE 36th International Conference on. IEEE, 2016, pp. 700–710.

[19] Z. Brakerski and V. Vaikuntanathan, “Fully homomorphic encryption from ring-lwe and security for key dependent messages,” in Annual Cryptology Conference. Springer, 2011, pp. 505–524.

[20] R. Bellafqira, G. Coatrieux, D. Bouslimi, and G. Quellec, “Content-based image retrieval in homomorphic encryption domain,” in 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, 2015, pp. 2944–2947.

[21] ——, “An end to end secure cbir over encrypted medical database,” in Engineering in Medicine and Biology Society (EMBC), 2016 IEEE 38th Annual International Conference of the. IEEE, 2016, pp. 2537–2540.

[22] P. Paillier, “Public-key cryptosystems based on composite degree residuosity classes,” in International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 1999, pp. 223–238.

[23] M. Nassar, A. Erradi, and Q. M. Malluh, “Paillier’s encryption: Implementation and cloud applications,” in Applied Research in Computer Science and Engineering (ICAR), 2015 International Conference on. IEEE, 2015, pp. 1–5.

[24] A. C. Yao, “Protocols for secure computations,” Foundations of Computer Science, 1982. SFCS ’82. 23rd Annual Symposium on, pp. 160–164, 1982.

[25] T. Vuegen, “Improving the dgk comparison protocol,” IEEE International Workshop on Information Forensics and Security (WIFS), pp. 49–54, 2012.
[26] S. G. Mallat, “A theory for multiresolution signal decomposition: the wavelet representation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[27] P. Zheng and J. Huang, “Discrete wavelet transform and data expansion reduction in homomorphic encrypted domain,” *IEEE Transactions on Image Processing*, vol. 22, no. 6, pp. 2455–2468, 2013.

[28] W. Gasarch, “A survey on private information retrieval,” *Bulletin of the EATCS*, 2004.