Secret and public key authentication watermarking schemes that resist vector quantization attack

Ping Wah Wong[a], Nasir Memon[b]

[a]Hewlett Packard Company, Cupertino, CA 95014
[b]Polytechnic University, Brooklyn, NY 11201

ABSTRACT

In previous work, Wong had proposed both secret key[10] and public key[11] watermarking schemes for image authentication that can detect and localized any change made to a watermarked image. The techniques proposed were block-based, that is, they partitioned the image into non-overlapping blocks and separately authenticated each block. Subsequently, Holliman and Memon[12] observed that many block based watermarking schemes are vulnerable to substitution attacks. They specifically showed that the Wong schemes can be attacked using a "vector quantization" (VQ) approach. This attack exploits that fact that if a sufficient number of images containing the same watermark bitmap is available, then one can use a VQ-like technique to forge a watermark into a new image. About the same time and independently, Coppersmith et al.[13] proposed to use overlapping blocks to resist this attack. Although this method can make the attack inefficient, it does so with a significant loss of the localization property of the watermark. We extend in this paper the Wong schemes[10,11] so that the resulting algorithms can resist the VQ attack and at the same time provide the same localization property in the watermark as the original schemes. The key idea is to insert a unique image-dependent block ID into the watermarking process so that the VQ attack will not have a rich enough "codebook" to forge the watermark.

1. INTRODUCTION

The problem for image authentication arises because digital images can be altered or manipulated with ease. In practice, it is generally impossible to tell whether a given image is authentic or has been altered subsequent to capture by some readily available digital image processing tools. This is an important issue in, for example, legal applications, news reporting, and medical archiving, where we want to be sure that the digital image in question truly reflects what the scene looked like at the time of capture. Another need of image authentication arises in, for example, electronic commerce where the seller transmits a digital image to the buyer over the network. In this case the buyer wants to be sure that the received image is indeed genuine. Here we not only want to verify the integrity of an image, but we also want to check for original ownership.

To address these issues, the idea of a trusted digital camera was proposed by Friedman.[2] Friedman's scheme computes a standard cryptographic signature for each captured image, which is stored and transmitted along with the image. Because a cryptographic signature is used, we not only can verify that the image has not been tampered with, we can also identify the origin of the image. That is, we can identify that the image was indeed captured by a specific camera. Recently, Yeung and Mintzer[3] proposed using a pseudo random sequence and a modified error diffusion method to embed a binary watermark to an image, so that any change in pixel values to the image can be detected. Extensions of this technique to compressed images were proposed by Wu and Liu.[14] It has been shown that the Yeung-Mintzer scheme is susceptible to substitution[12,4] and key-recovery[5] attacks under certain conditions. Lin and Chang[6] proposed a scheme to insert authentication data in JPEG coefficients so that the authentication watermark has some resilience against JPEG compression. They later extended the scheme to handle video data.[7]

Wong proposed a secret key watermarking scheme[10] and a public key watermarking scheme[11] for image authentication. These schemes allow a user with an appropriate key to verify the integrity and the ownership of an image. Furthermore, this authentication watermark can detect and localize any change to the image, including changes in pixel values or image size. The security of these two schemes resides in the secrecy of the user key and not in the obscurity of the algorithm. In fact, the watermark insertion and extraction steps can be made public without compromising the security of the watermark.

Later, Holliman and Memon[12] discovered that the schemes in,[10,11] as well as many other block-based watermarking schemes proposed in the literature were vulnerable to a substitution attack. Particularly, they showed that the

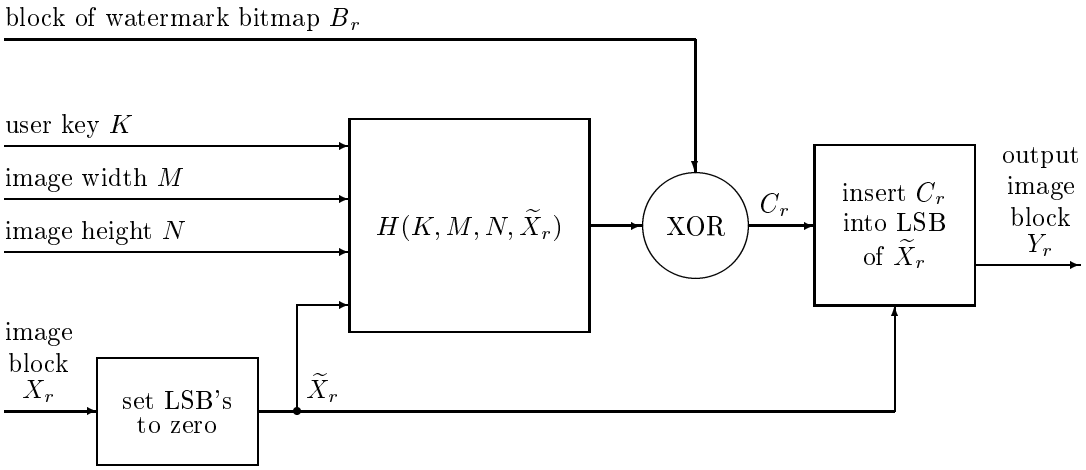

Figure 1. Secret key authentication algorithm.[10] Inserting a secret key authentication watermark to each image block. The image is partitioned into blocks, each of size I times J pixels, and the watermark insertion procedure is applied independently to each block of the image. This scheme is susceptible to the Holliman Memon VQ attack.[12]

two Wong schemes were vulnerable to a "vector quantization" attack. That is, they showed that given a sufficient large number of images watermarked using the same watermark bitmap, one can forge a watermark for a new image without knowing the watermark insertion key. In 1999, Coppersmith et al.[13] proposed an authentication technique very similar to the public key scheme in,[11] except that Coppersmith et al. used overlapping blocks to effectively defeat the "vector quantization" attack. However, one disadvantage of using overlapping blocks is that the localization property of the watermark (the ability to localize changes to the watermarked image) is substantially reduced relative to the block size. In this paper, we introduce an index parameter in a hash function for the Wong schemes that makes the "vector quantization" attack practically infeasible, and at the same time preserves the localization properties of the watermark in.[10,11]

In Section 2, we briefly review the secret key and public key schemes proposed previously by Wong.[10,11] Section 3 briefly describes the "vector quantization" attack developed by Holliman and Memon. Section 4 discusses the modification to the previous secret key and public key schemes, and their resiliency to the VQ attack. Section 5 summarizes the results of this paper.

2. SECRET KEY AND PUBLIC KEY AUTHENTICATION WATERMARKS

In this section we review the secret key and public key authentication watermarking algorithms that were proposed by Wong.[10,11] For ease of exposition, we describe the technique in reference to greyscale images, just as was done in.[10,11] For color images, the same technique can be applied independently to the color planes of the image, either in the RGB color space or in any other color space such as YUV.

2.1. Secret Key Authentication Watermarking Algorithm

Consider a greyscale image $x_{m,n}$ of size M_X by N_X pixels. To insert an invisible watermark to form a watermarked image $x^w_{m,n}$ of the same size, we first partition $x_{m,n}$ into blocks of $I \times J$ pixels. The scheme then inserts an invisible watermark to each block of image data. The watermark insertion procedure for each image block is shown in 1.

Let $a_{m,n}$ be a bi-level image that we will use as our invisible watermark, to be embedded in $x_{m,n}$. Note that $a_{m,n}$ needs not be of the same size as $x_{m,n}$. From $a_{m,n}$, we form another bi-level image $b_{m,n}$ of size $M_X \times N_X$ (same size as $x_{m,n}$). There are many ways of doing so. In our example, we form $b_{m,n}$ by tiling $a_{m,n}$, i.e., periodically replicating $a_{m,n}$ to the desired size. Another possibility, provided the difference in size between $x_{m,n}$ and $a_{m,n}$ is small, is to append all zeros (or all ones) to the boundary of $a_{m,n}$ so that we obtain $b_{m,n}$ of the desired size.

Let

$$X_r = \{x_{iI+k,jJ+l} : 0 \le k \le I-1; 0 \le l \le J-1\}$$

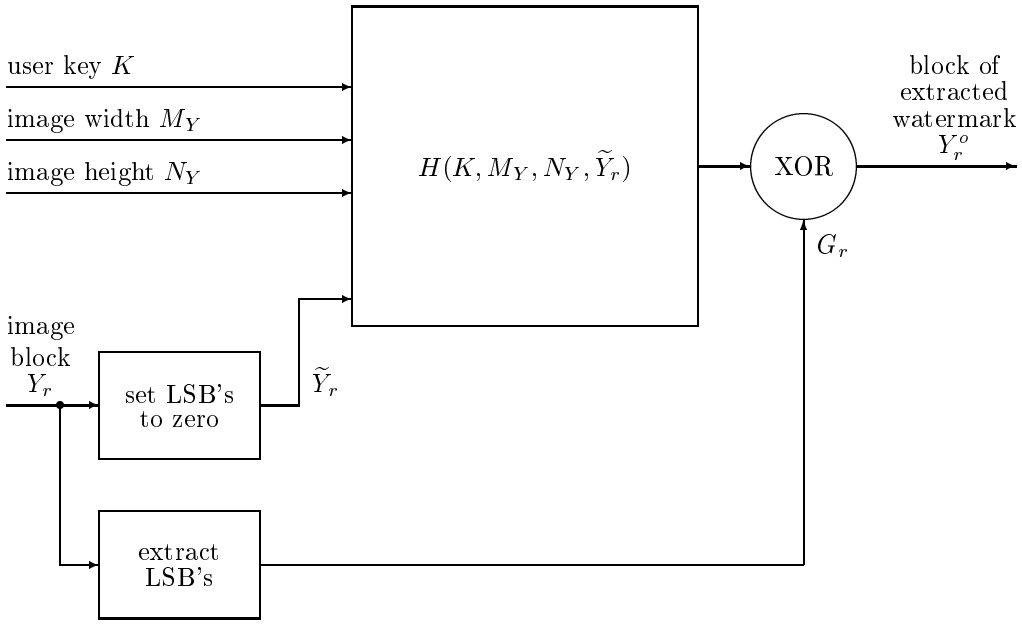

Figure 2. Block diagram of the extraction procedure for the invisible watermark for each image block.

be a block of size $I \times J$ taken from the image $x_{m,n}$. For simplicity, we use a single index r to denote the r^{th} block in the image. The corresponding block within the binary image $b_{m,n}$ is denoted by

$$B_r = \{b_{iI+k,jJ+l} : 0 \leq k \leq I-1; 0 \leq l \leq J-1\}.$$

Consider a cryptographic hash function

$$H(S) = (d_1, d_2, \ldots, d_p)$$

where S represents a string of data of arbitrary length, d_i's are the binary output bits of the hash function, and p is the size of the output bit string. A cryptographic hash function has the property that given an input bit string S and its corresponding output $(d_1, \ldots, d_p)$, it is computationally infeasible to find another input bit string of any length that will be hashed to the same output $(d_1, \ldots, d_p)$. An example is the well known MD5[8] where any string of data is hashed into a bit array of length 128, i.e., $p = 128$. For the rest of this paper, we will use MD5 as our hash function. It is obvious that any other cryptographic hash function can also be used in our watermarking scheme.

Let K be a user key consisting of a string of bits. For each block of data X_r, we form the corresponding block $\widetilde{X}_r$ where each element in $\widetilde{X}_r$ equals the corresponding element in X_r except that the least significant bit is set to zero. We compute for each block the hash

$$H(K, M_X, N_X, \widetilde{X}_r) = (d_1^r, d_2^r, \ldots, d_p^r). \tag{1}$$

In the case where $p < I \times J$, We extend the p bits that constitute the hash output by replication and form the rectangular array $d_{m,n}$ of size $I \times J$. Otherwise* if $p \geq I \times J$ we take the first $I \times J$ bits to form $d_{m,n}$. The array $d_{m,n}$ is then combined with B_r to form a new binary block C_r using a pixel by pixel exclusive OR operation. That is, we form

$$c_{m,n} = b_{m,n} \oplus d_i,$$

where $\oplus$ is the exclusive OR operation, and $c_{m,n}$ are the elements in C_r. Finally we put $c_{m,n}$ into the least significant bit of the block $\widetilde{X}_r$ to form the output watermarked image block X_r^w. This procedure is repeated for each block of data, and all the output blocks X_r^w are assembled together to form the watermarked image $x_{m,n}^w$.

*As explained in,[10,11] it is highly desirable $p \leq I \times J$.

Figure 3. An original image.

Figure 4. An invisible watermark added to the image of Fig. 3.

The extraction procedure for the invisible watermark is shown in 2. The least significant bits of each element in the block Y_r is set to zero to give a block $\widetilde{Y}_r$. For each block of data Y_r, we compute $H(K, M_Y, N_Y, \widetilde{Y}_r)$ and perform a pixel by pixel exclusive OR operation with G_r to form a block of the output binary watermark.

Consider the case where the watermarked image was not changed. That is, we have $X_r^w = Y^r, M_X = M_Y$ and $N_X = N_Y$. This implies $\widetilde{Y}_r = \widetilde{X}_r$ and $C_r = G_r$. Hence we have $Y_r^o = B_r$. That is, the appropriate watermark bitmap block is extracted. If the image is changed in any way, either by changing the pixel value or changing the size, or if the inappropriate key is used in the extraction algorithm, then the extract watermark bitmap will not be correct. Because of the properties of the hash function, that the output digest will vary "wildly" for any change to the input, the extracted watermark bitmap actually resembles random noise if the input is changed.

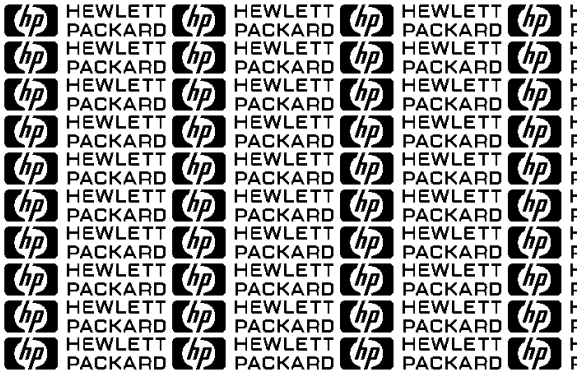

Figure 5. Extracted watermark from the image in Fig. 4.

Figure 6. Extracted watermark from the image in Fig. 4 using an incorrect user key. The output resembles random noise. Similar output will result if the image contains no watermark, of if a watermarked image is cropped.

Figure 7. The watermarked image in Fig. 4 has been changed. A glass was pasted on top.

Figure 8. Extracted watermark from Fig. 7 indicating the location where changes have occurred.

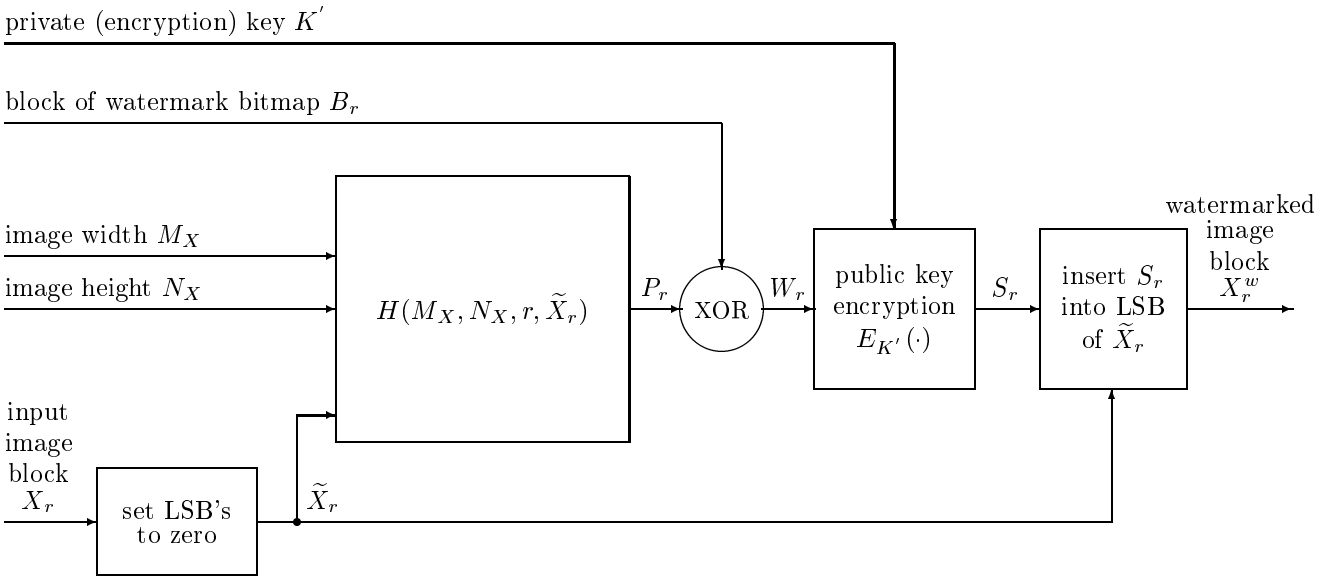

Figure 9. Public key verification watermark insertion procedure.

The invisible watermarking technique described above allows the detection of any change to a watermarked image. It can also allow ownership verification through a secret key K which is used to embed the watermark and which is known only to the the owner and verifier. Properties of the technique and detailed discussion on its security are given in.[10] Figure 3 shows an original image and Figure 4 shows an image with an invisible watermark added using the technique described above. If one uses the correct user key K and applies the watermark extraction procedure to Figure 4, one obtains an output image Figure 5, indicating the presence of a proper watermark. If an image is unmarked,or if one applies an incorrect key (for example, if one does not know the key), or if a watermarked image is cropped then the watermark extraction procedure returns an output that resembles random noise as shown in Figure 6. On the other hand, if one changes certain pixels in the watermarked image, then the specific locations of the changes are reflected at the output of the watermark extraction procedure. Figure 7 shows an image where a glass is pasted onto Figure 4. Figure 8 shows the extracted watermark from Figure 7, indicating the specific area where changes have been made. This property, of localizing pixel modifications can be highly desirable in some applications. But it is this very feature that causes to weakness exploited by Holliman and Memon as described in the next section. This leads to the question whether one can resist the Holliman-Memon attack and at the same time retain the localization property illustrated in Figures 8 and 7? We show in section 4 that this is indeed possible by a minor modification to the above technique.

2.2. Public Key Authentication/Verification Watermark

The previous section deals with a secret key watermarking method for authentication purposes. Being a secret key scheme, the same key is used for both watermark insertion and extraction. Hence, the key must be transmitted from the owner to the verifier through a secure channel. In this section, we extend the secret key watermarking method so that the integrity and ownership of the image can be verified using a public key. In such a system, the owner of the image inserts a watermark using a private key $K^{'}$. In the watermark extraction procedure, any person can use the public key K (corresponding to the private key $K^{'}$) to extract a binary watermark, that will indicate changes, if any, that have been made to the watermarked image.

As before, want to insert a binary watermark image $b_{m,n}$ to $x_{m,n}$ to obtain the watermarked image $x^w_{m,n}$. The public-key watermark insertion and extraction procedures are shown in Figures 9 and 10, respectively.

Let X_r denote the r^{th} block of data within the image $x_{m,n}$. Just as in the secret scheme described in the previous section, we form the corresponding block $\widetilde{X}_r$ where each element in $\widetilde{X}_r$ equals the corresponding element in X_r except that the least significant bit is set to zero. Let $H(\cdot)$ be a cryptographic hash function such as the MD5.[8] We compute the hash

$$H(M_X, N_X, \widetilde{X}_r) = (p_1^r, p_2^r, \ldots, p_s^r) \tag{2}$$

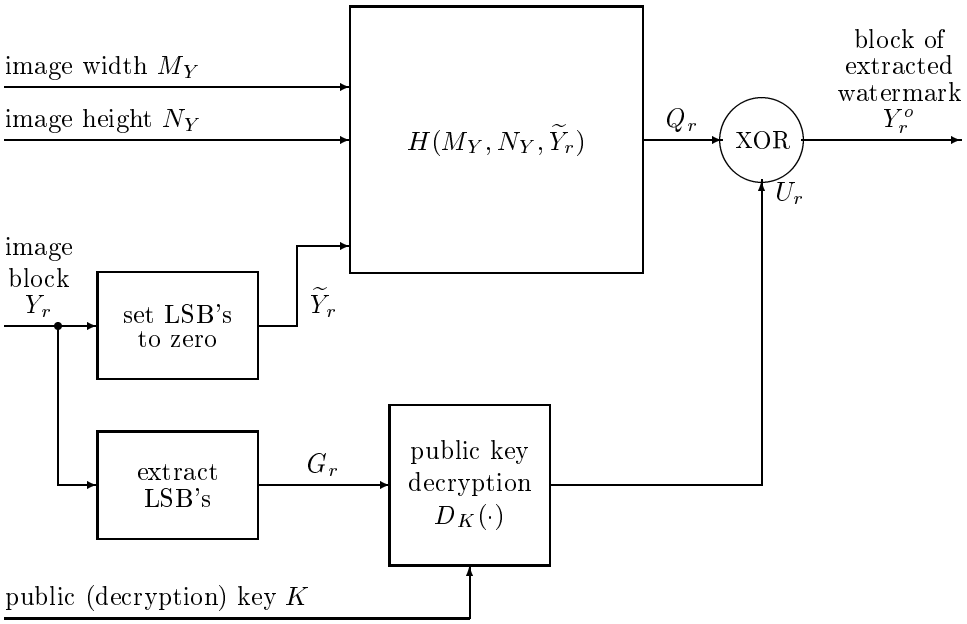

Figure 10. Public key verification watermark extraction procedure.

where p_i^r denotes the output bits from the hash function, and s is size of the output bits that is dependent on the specific hash function used. For example, $s = 128$ for MD5. Denote these 128 bits suitably extended (say by replication), to the block size $I \times J$ (144 bits when the block size is 12×12) as P_r, i.e.,

$$P_r \triangleq (p_1^r, p_2^r, \ldots, p_{I\times J}^r).$$

We combine P_r with a corresponding block B_r in $b_{m,n}$ using an exclusive or function. That is, we compute $W_r = P_r \oplus B_r$ where $\oplus$ denotes the element-wise exclusive OR operation between the two blocks. Finally we encrypt W_r with a public key cryptographic system[9] to give

$$S_r = E_{K'}(W_r)$$

where $E(\cdot)$ is the encryption function of the public key system, and $K^{'}$ is the private key. The binary block of data S_r is then embedded into the least significant bit of the data block $\widetilde{X}_r$ to form a block X_r^w in the watermarked image.

In the extraction procedure, we split the input image block Z_r into two pieces; the first piece G_r contains the least significant bits, and the other piece $\widetilde{Z}_r$ contains the pixel values except that the least significant bits have been zeroed out. We then calculate the hash of M, N and $\widetilde{Z}_r$, and denote the first $I \times J$ bits of the output by Q_r. We use a public key decryption algorithm[9] to decrypt G_r with the *public key* K that corresponds to the private key $K^{'}$ used in the watermark insertion procedure. That is, we calculate

$$U_r = D(Z_r).$$

Finally, we compute the output block $O_r = Q_r \oplus P_r$ using an element-wise exclusive or procedure.

In our implementation of the public key watermark insertion and extraction procedures, we used the MD5[8] as our hash function, and the RSA public key encryption algorithm[9] for encryption and decryption. It is evident that any publikc key encryption/decryption and any cryptographic has function can be used to achieve similar properties for the watermarking algorithm.

Again, the properties of this technique are similar to the private key based technique described earlier and are discussed in detail in[11] and summarized in Figure 11. We would like to highlight the fact that, as before, this algorithm can detect any change made to the pixel values to the block level.

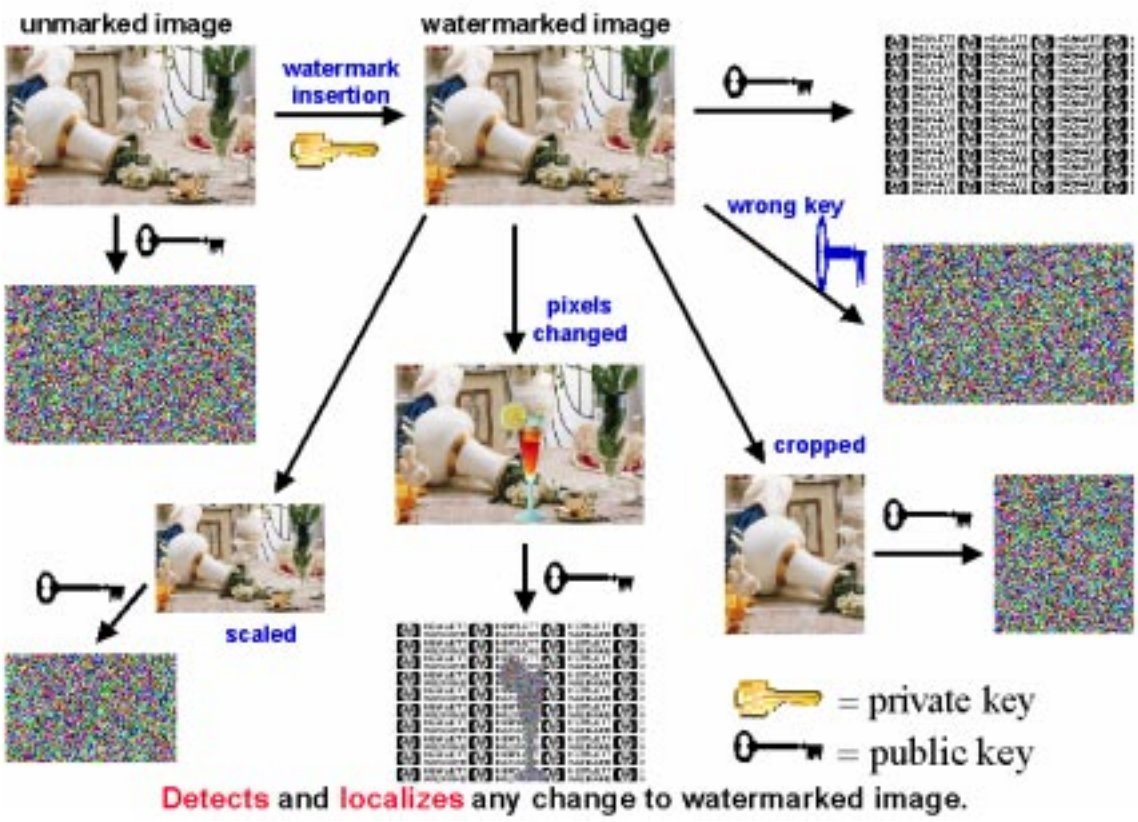

Figure 11. Experimental results summarizing the properties of the public key verification watermark.

3. THE HOLLIMAN-MEMON ATTACK

Both the secret key and public key authentication watermarking techniques described in the previous sections are vulnerable to a substitution attack as shown by Holliman and Memon.[12] We show in the next section how these two techniques can be modified to prevent this attack. However, before we do so we briefly describe here the attack by Holliman and Memon.

Suppose we have a set of watermarked images $\{v_{m,n}^{(1)}, v_{m,n}^{(2)}, \ldots, v_{m,n}^{(L)}\}$, all containing the same watermark bitmap (binary logo) $b_{m,n}$, and all are watermarked using the same key K (the secret key in the secret key watermarking algorithm, and the private key in the public key watermarking algorithm). We denote the blocks within each watermarked image by $\{V_r^{(1)}, V_r^{(2)}, \ldots, V_r^{(L)}\}$. Consider an image $u_{m,n}$ where the blocks are denoted by U_r. The Holliman-Memon attack is aimed at finding a "watermarked image" $v_{m,n}$ so that if the watermarking extraction algorithm is applied to $v_{m,n}$ using the appropriate key, then we obtain the watermark bitmap $b_{m,n}$. In this attack, it is assumed that the key is *not* known to the attacker. It is also assumed that the watermark bitmap is known. The assumption that the attacker would know the watermark bitmap is not restrictive particularly in the public key scheme.

The Holliman-Memon attack works as follows: First the attacker would sort the blocks in the L watermarked images so that all the blocks in the collection $\{V_r^{(1)}, V_r^{(2)}, \ldots, V_r^{(L)} r := 1, 2, \ldots\}$ that correspond to the same watermark bitmap block are collected. The attack proceeds by examining each block U_r of the image $u_{m,n}$. For each block U_r, the attacker identifies the necessary output block of the watermark bitmap. Then from the subset of the blocks $\{V_r^{(1)}, V_r^{(2)}, \ldots, V_r^{(L)} r := 1, 2, \ldots\}$ that would give the same output watermark bitmap block, the attacker finds the block that minimizes the distortion between the block and U_r. This is essentially the same as vector quantization, where we think of a "code book" as the collection of all the blocks that would be "decoded" into the same watermark bitmap.

4. MODIFICATIONS TO PREVENT THE HOLLIMAN-MEMON ATTACK

One factor which affects the ease by which the Holliman-Memon counterfeiting attack can be carried out is the nature of the watermark logo image, or in other words the number of VQ codebooks needed. In the worst case scenario, from the counterfeiter's point of view, if every $k \times l$ block in the watermark logo is unique then the number of codebooks that need to be created is $\frac{m \times n}{k \times l}$, that is, one codebook for each block position in the $m \times n$ image. For a block size of 8×8, in order to construct a reasonable quality counterfeit watermarked image, an attacker might require, say, 128 entries in each codebook–that is, only 128 watermarked images, even for this worst-case scenario. Such a figure is easily attainable, particularly given a large archive of watermarked images which are similar, like a fingerprint database. A tiled logo image could easily result in an even smaller number of watermarked images being required for successful counterfeiting.

Another factor that has a considerable impact on the counterfeit watermarked image quality is the size of each image block. All other factors aside, it's obviously easier to construct a reasonable approximation to a smaller block than a larger one. So, one potential solution to the problem is to simply increase the block size used by a given watermarking algorithm. However, a small block size is desirable for accurate localization of image changes. For these reasons, we believe that increasing the block size does not represent a good approach for thwarting the proposed counterfeiting attacks.

A third solution is to add the block index r to the input of the hash function block. Then for each block U_r to be forged, the attacker is constrained to consider only blocks from all the images that has the same index r. Hence, the size of the code book is necessarily small, i.e., one entry in the codebook per watermarked image in the collection. But then this is the same as the first solution proposed in this section, that is, using a watermark logo with unique blocks. As we have already argued, this may not be sufficient when dealing with a large database of images, all watermarked with the same key and same logo.

Yet another clever and effective approach to thwart the Holliman-Memon attack proposed independently in,[13] suggests using a larger surrounding neighborhood (e.g. 16×16) to be hashed and inserted in a smaller (say 8×8) block at location (m, n). For a sufficiently large neighborhood, this is an elegant solution to the problem. However the approach of Coppersmith et al. comes at the cost of reduced error localization properties.

A further way to completely thwart the Holliman-Memon attack is to add an image index $\mathcal{I}$ to the input of the hash function. Here the index is unique to each image, and serves the purpose as the serial number of the image. In this way, the Holliman-Memon attack cannot take place. At first sight it may appear that the disadvantage to requiring $\mathcal{I}$ be included as input to the hash function is that the verifier who wants to check the image must be informed of the exact ID of the image. Now, in certain applications, such as a merchant who sells images on a public web site, this may not be a restrictive requirement because the image ID information can be placed publicly on the web site. On more careful examination, given the nature of the techniques proposed by Wong in,[10,11] $\mathcal{I}$ can be computed from the MSB planes of the image X itself! For example, it could just be the hash value of all the $\widetilde{X}_r$ taken together. This is possible as the Wong image authentication techniques[10,11] only insert the watermark in the LSB and do not affect higher order bits.

So in the modified techniques, we compute for each block the hash

$$H(K, \mathcal{I}_X, M_X, N_X, r, \widetilde{X}_r) = (d_1^r, d_2^r, \ldots, d_p^r).$$

Note that the two parameters $\mathcal{I}_X$ and r are used in order to resist the "vector quantization" attack proposed by Holliman and Memon.[12] Figures 12 and 13 show the Wong authentication techniques with the proposed modifications. The corresponding watermark extraction techniques can be similarly modified and are omitted for brevity.

5. CONCLUSION

We have described modifications of the original Wong watermarking scheme for ownership verification and authentication. The modifications, although small, make the Holliman-Memon attack ineffective and yet the same time preserve all the desirable properties of the original techniques. Finally, we would like to note that although we have described our authentication watermarking techniques for uncompressed images in the spatial domain, they can be extended to compressed images in the transform domain by using techniques similar to those proposed by Wu and Liu in.[14]

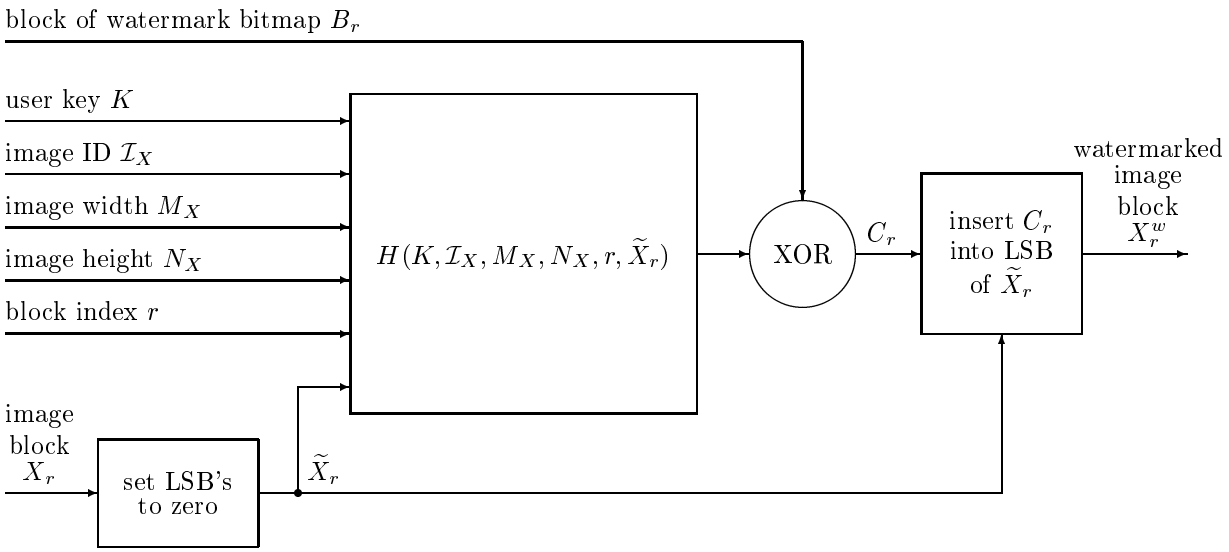

Figure 12. Modified secret key authentication algorithm. This scheme prevents the Holliman Memon VQ attack

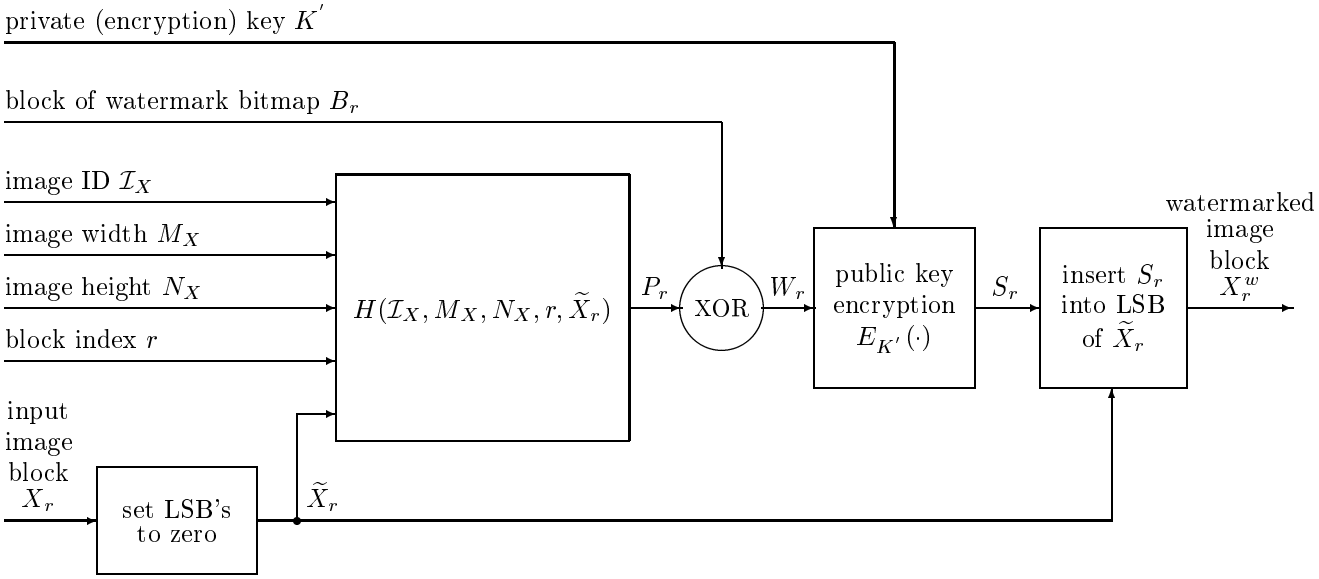

Figure 13. Modified public key verification watermark insertion procedure. This technique resists the Holliman-Memon attack

REFERENCES

1. F. Mintzer, G. Braudaway, and M. M. Yeung, "Effective and ineffective digital watermarks," in *Proceedings of ICIP*, (Santa Barbara, CA), October 1997.
2. G. L. Friedman, "The trustworthy digital camera: restoring credibility to the photographic image," *IEEE Transactions on Consumer Electronics*, vol. 39, pp. 905–910, November 1993.
3. M. M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification," in *Proceedings of ICIP*, (Santa Barbara, CA), October 1997.
4. Jiri Fridrich, Miroslav Goljan, and Nasir Memon. "Further Attacks on Yeung-Mintzer Fragile Watermarking Scheme", in *Proceedings of SPIE/IS&T Electronic Imaging Symposium, Security and Watermarking of Multimedia Contents*, (San Jose, CA), January 2000.
5. N. Memon, S. Shende and P. Wong, "On the security of the Yeung-Mintzer fragile watermarking technique," in *Proceedings of SPIE*, To be fixed, Savannah, GA, March 1999.
6. C.-Y. Lin and S.-F. Chang, "A robust image authentication method surviving JPEG lossy compression," in *Proceedings of SPIE, vol. 3312*, pp. 296–307, 1998.
7. C.-Y. Lin and S.-F. Chang, "Issues and solutions for authenticating MPEG video," in *Proceedings of SPIE/IS&T Symposium on Electronic Imaging, Security and Watermarking of Multimedia Contents*, (San Jose, CA), January 1999.
8. R. L. Rivest, "The MD5 message digest algorithm." Internet RFC 1321, April 1992.
9. R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, pp. 120–126, February 1978.
10. P. W. Wong, "A watermark for image integrity and ownership verification," in *Proceedings of IS&T PIC Conference*, (Portland, OR), May 1998. Also available as Hewlett Packard Laboratories Technical Report HPL-97-72, May 1997.
11. P. W. Wong, "A public key watermark for image verification and authentication," in *Proceedings of ICIP*, (Chicago, IL), October 1998.
12. M. Holliman and N. Memon, "Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes," *IEEE Transactions on Image Processing*, To appear March 2000.
13. D. Coppersmith, F. Mintzer, C. Tresser, C. W. Wu, and M. M. Yeung, "Fragile imperceptible digital watermark with privacy control," in *Proceedings of SPIE/IS&T Electronic Imaging Symposium, Security and Watermarking of Multimedia Contents*, (San Jose, CA), January 1999.
14. M. Wu and B. Liu, "Watermarking for Image Authentication", in *Proceedings of ICIP*, (Chicago, IL), October 1998.