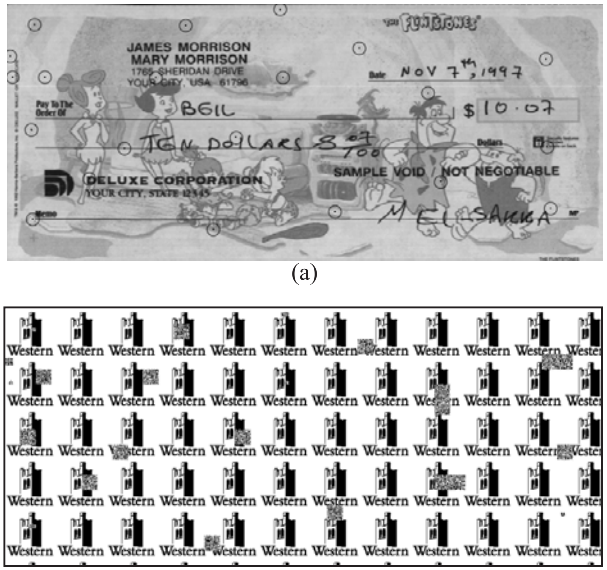
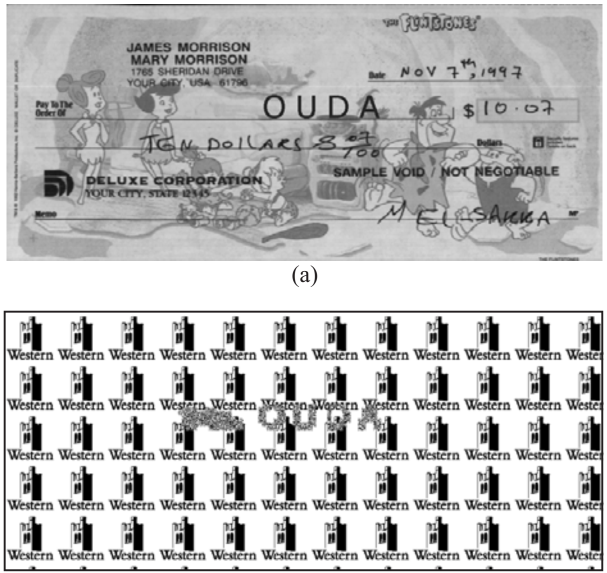


(a)

(b)

Figure 2: (a) The watermarked image with some random modifications, where the RSA-309 is used.
(b) The corresponding extracted binary image.



(a)

(b)

Figure 3: (a) The modified watermarked image with unchanged LSBs, where the RSA-309 is used.
(b) The corresponding extracted binary image.

6. CONCLUSION

This paper showed that, Wong's watermarking scheme suffers from a serious security leak. The main reason of this leak is that, the authors made a fundamental mistake of assuming that the plaintext size determines ciphertext size. We demonstrated in this paper that, it is the key size that determines ciphertext size. Therefore, if a small key is applied to produce a small ciphertext that can be embedded in the small image blocks, the scheme will be insecure. And if a long key is used, the Wong's scheme will always give misleading results. This paper has come up with an elegant solution to tackle this leak. A larger image block was recommended to hold the entire watermark. A new method of applying the cryptographic hash function MD5 is utilized to achieve a high-level of localization accuracy. This solution moved the Wong technique from a cryptographically insecure system to a reliable and secures one.

7. REFERENCES

[1] P. Wong, “A Public Key Watermark for Image Verification and Authentication,” IEEE International Conference on Image Processing (ICIP’98), vol. I, pp. 455–459, October 1998.

[2] N. Memon and P. Wong, “Secret and Public Key Authentication Watermarking Schemes that Resist Vector Quantization Attack,” SPIE International Conference on Security and Watermarking of Multimedia Contents, vol. 3971, pp. 471–427, January 2000.

[3] M. Holliman and N. Memon, “Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes,” IEEE Transactions on Image Processing, vol. 9, no. 3, pp. 432–441, 2000.

[4] P. Wong and N. Memon, “Secret and Public Key Image Watermarking Schemes for Image Authentication and Ownership Verification,” IEEE Transactions on Image Processing, vol. 10, no. 10, pp. 1593–1601, October 2001

[5] R. Rivest. “The MD5 message digest algorithm,” Technical Report RFC1321, IETF, 1992.

[6] R. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” Communications of the ACM, vol. 21, no. 2, pp. 120–126, 1978.

[7] R. Rivest, A. Shamir, and L. Adleman, “RSA challenge,” Scientific American, M. Gardner's column, 1977.

[8] D. Atkins, M. Graff, A. Lenstra, and P. Leyland, “The magic words are squeamish ossifrage,” Advances in Cryptology, LNCS 917, pp. 263–277, Springer, Berlin, 1995.

[9] S. Cavallar, B. Dodson, A. Lenstra, P. Leyland,W. Lioen, P. Montgomery, B. Murphy, H. Riele and P. Zimmermann, “Factorization of RSA-140 using the Number Field Sieve,” Advances in Cryptology, LNCS 1716, pp. 195–207, Springer, Berlin, 1999.

[10] S. Cavallar, W. Lioen, H. Riele, B. Dodson, A.Lenstra, P. Montgomery and B. Murphy, “Factorization of a 512-bit RSA Modulus,” Advances in Cryptology, LNCS 1807, pp. 1–8, Springer, Berlin, 2000.

[11] P. Flajolet, D. Gardy and L. Thimonier, “Birthday paradox, coupon collectors, caching algorithms and self-organizing search,” Discrete Applied Mathematics, vol. 39, no. 3, pp. 207–229, 1992.

provides an acceptable level of security. This assumption is based on the following two reasons.

First: A collision may be naturally occurred, i.e., two different image blocks were mapped into one hash output. Based on the birthday paradox [11], given a hash function that produces a n-bit long output, it is expected that after trying $2^{n/2}$ possible input values, the collision should happen. In this case the probability that two different image blocks are mapped to same 64 bits of the MD5 hash output is $1/2^{32}$. This probability is quite acceptable compared with the total number of blocks in a given image.

To support this claim, an experiment was made to test the collision resistance of the first 64 bits of the MD5 output. A database of 650 digital images was used. Each image was 1274×552 pixels, which in turn was divided into 8×8 pixel blocks. Note that, for simplicity the largest area of the image that is a multiple of 8×8 is considered, (i.e., the border conditions are ignored). Then MD5 is applied on each block, the result showed that no two outputs shared the same first 64 bits. In fact, we found that the first 46 bits were long enough to distinguish any two outputs.

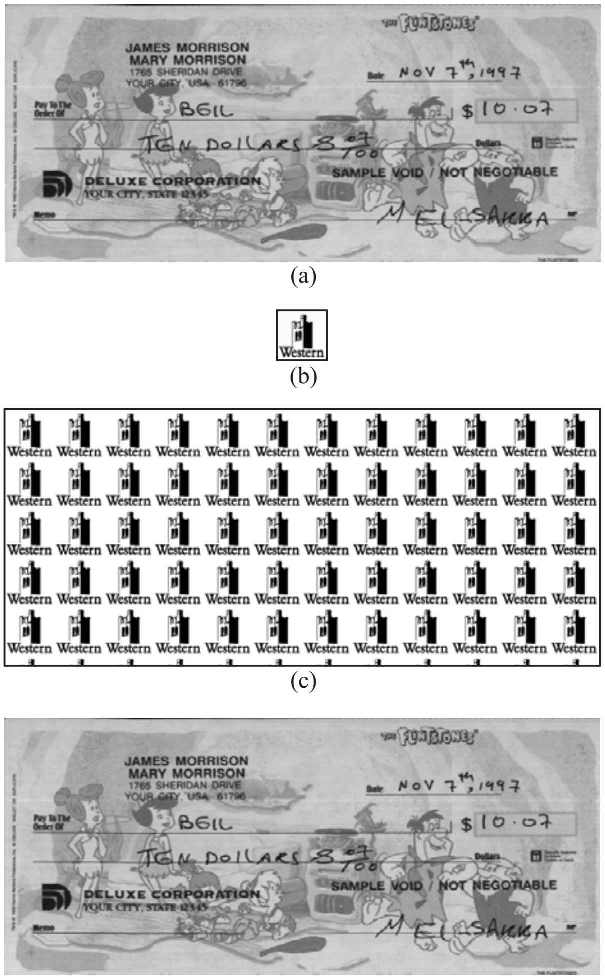
Second: The goal of the attacker is to make changes in the watermarked image that may be undetected in the verification process. This means that the modified image block(s), which are meaningful to the attacker, should produce a hash value exactly as that produced by these blocks. Typically, this is much harder to achieve over the collision problem mentioned above.

5. EXPERIMENTAL RESULTS

To validate the proposed algorithm, the following experiment has been made, where the image shown in Figure 1(a) is used as an unmarked image and the image in Figure 1(b) is used as the input binary logo image. The image in Figure 1(c) shows the binary watermarking image, which was produced from tiling the binary logo image. The produced watermarked image is shown in Figure 1(d). The verification algorithm is tested on three different cases:

Case 1: No modifications were made on the watermarked image. The extracted binary image, after applying the verification algorithm on the watermarked image, is compared bit by bit with the image in Figure 1(c). It is found that both images are exactly the same. In practice, this means that the encryption and decryption processes were successfully performed, and the image block size used (i.e., 32×32) was large enough to hold the entire watermark (i.e., the ciphertext).

Case 2: The watermarked image was modified such that arbitrary numbers of black dots (19 dots in total) were added in different places in the image. Note that, these dots are 3×3 pixels to be easily visualized. The number and the locations of these dots were randomly chosen, see Figure 2(a). Note that, these modifications were made without regards to whether the LSBs of these areas are altered or not. Note also that, the circles around these dots are placed for clarification purposes only and they are not part of the modified image.



(a)

(b)

(c)

(d)

Figure 1: (a) The original image, (b) The binary logo, (c) The binary image and (d) The watermarked image.

After applying the verification process, the produced binary image is shown in Figure 2(b). It is clear from the image that the modified areas have been successfully identified. The dimensions of the identified blocks vary from 8×8 pixels to 32×32 pixels. Typically, the scrambled 8×8 pixel blocks appear when the LSBs of the modified areas have not been changed and hence the extracted and calculated hash values of some 8×8 image block may be identical. On the other hand if any of the LSBs were modified, then a scrambled 32×32 pixel block will appear instead.

Case 3: The watermarked image was modified in some areas such that the LSBs of their pixels were not changed, see Figure 3(a), where the payee name is scratched out and another name was added. By applying the verification procedure on the watermarked image after these modifications, the extracted binary image is shown in Figure 3(b). Note that, the modified areas have been successfully identified, and their locations are sufficiently detected. This accuracy is due to the fact that the LSBs of the image were kept untouched.

Wong's technique is presented. Whereas, the main problem of this technique is described in Section 3. Section 4 explains how this problem can be solved. The experimental results that support this solution is shown in Section 5. Finally, the conclusion is offered in Section 6.

2. WONG'S WATERMARKING TECHNIQUE

Wong's technique works by inserting a watermark image $b_{m,n}$ into an image $x_{m,n}$ where both of them are of size $m\times n$. The watermark $b_{m,n}$ is a binary image, which can be generated by tiling another smaller binary image. The main purpose for this insertion is, if there is any change made to the image, the extracted watermark $b'_{m,n}$ should be affected and hence locating the effected area(s). Note that, both the image and the watermark are partitioned into a number of small blocks of sizes $i\times j$ pixels, and then they are embedded on the corresponding image blocks.

The following algorithm is used to embed the binary watermark into the image: (note that the image $x_{m,n}$ and the watermark $b_{m,n}$ will be divided into small blocks X and B respectively)

For each block X in the data image do:

1. Generate a block $\overline{X}$ which is exactly the same as the block except that the least significant bits (LSBs) are set to zero
2. Apply a cryptographic hash function H to compute the hash
$$H(m,n,\overline{X})=(p_1,p_2,\cdots,p_s) \tag{3}$$
where p_i denotes the output of the hash function in bits, and s is its length. Based on Wong's recommendations, MD5 [5] will be used as a hash function, and hence s will be 128. The author mentioned that the image block size ($i\times j$) should be less than or equal to s . Hence, let P be a stream of the first $i\times j$ bits from the output of Equation (3), i.e.,
$$P=(p_1,p_2,\cdots,p_{ij}) \tag{4}$$
3. Combine P with the corresponding watermark image block B , using the exclusive OR operation to obtain the data block W , i.e.,
$$W=P\oplus B \tag{5}$$
4. Use the public-key Encryption RSA to encrypt *W*, to produce the ciphertext C , i.e.,
$$C=E_{private}(W) \tag{6}$$
where $E_{private}$ is the RSA encryption function. Note that, Equation (6) can be re-written as:
$$C=(W^{d}\ ,\text{mod}\, n) \tag{7}$$
where d is the RSA private key, see Section 1.
5. Embed the binary representation of C into the LSBs of the image block $\overline{X}$.

At the receiving end, for each image block the steps 1 and 2 are repeated to get P' . At the same time the ciphertext C' is extracted from the corresponding image block. The matched public key of the RSA is applied on C' to get W' . And the block watermark is then computed by $B'=P'\oplus W'$.

After scanning all of the image blocks, the authenticity of image can be visually detected by comparing the original watermark $b_{m,n}$ with the extracted watermark $b'_{m,n}$.

3. THE MAJOR PROBLEM

This technique has the ability to detect small changes made by a malicious attack and sufficiently identify the location of these modifications. This is true because the block sizes are chosen to be relatively small (8x8 pixels) as mentioned in [1].

The length of the RSA keys used in the Wong technique was not mentioned in the original paper, but it should be one of the following two options:

<u>Option 1</u>: Assume that the key is chosen to be safe, see Section 1, that is the key length should not be less than 768 bits (the length of *n*) i.e., RSA-232 is used. Note that the length of the RSA-232 output is seldom below 64 bits. This means that the binary representation of C , see Equation (7), will almost always truncated before its insertion on the image block. This is because it will take the first 64 bits out of 768 bits.

In the extraction and detection process, the decrypted W' will almost always differ than its original W . This means that the Wong's algorithm will always give incorrect authentication results.

<u>Option 2</u>:To guarantee that the ciphertext length will not exceed 64 bits, it seems that RSA-20 (64-bit modulus) is the most suitable RSA that can be used in Wong's technique. In this case Wong's algorithm always gives correct results, but it uses a short keys, i.e., insecure cryptosystem.

Therefore, in both options the Wong technique either gives misleading results or its security can be easily broken.

4. PROPOSED MODIFICATION

One choice that may push Wong's watermarking technique far from the hands of cryptanalysts is the use of longer key lengths. Our advice is to upgrade the key length to 1024 bit encryption, i.e., RSA-309. This will ensure that it stays ahead of the code breakers, especially after Cavallar's announcement in [10] that "within 7 years from now the 768-bit (232-digit) RSA keys will become unsafe". Therefore, in order to embed the entire ciphertext, i.e., 1024 bits, into the image block, larger block sizes are needed. In this case the image should be divided into 32×32 pixel block sizes instead of 8x8 pixel blocks. But our aim is also to achieve good level of detection accuracy, therefore each 32×32 pixel block is subdivided into 16 non-overlapping blocks, each of them of size 8×8 pixels. The MD5 algorithm is applied individually on these blocks and then the first 64 bits of each output is concatenated to form a hash stream for the 32×32 pixel block. This stream is of length 1024 bits. Before we show the proposed algorithms, let us first describe how the first 64-bits in the MD5 hash stream can be safely used.

MD5 is a one-way function that takes a variable-size input string and returns a fixed-length string of size 128 bits. It is assumed that it is computationally infeasible to find two input strings having the same output, or to find any input string for a given pre-specified output. It is worth mentioning that considering the first 64 bits from the output of MD5 still

A PRACTICAL VERSION OF WONG'S WATERMARKING TECHNIQUE

Abdelkader H. Ouda and Mahmoud R. El-Sakka, Senior Member IEEE

Computer Science Department, University of Western Ontario,
London, Ontario, N6A 5B7, Canada
{kader, elsakka}@csd.uwo.ca

ABSTRACT

In this paper, we study the security of Wong's technique. It is shown that the technique is vulnerable to cryptographer's attacks. This is due to the use of short keys in the public-key cryptosystem. Short keys are used in Wong's technique to make the watermark small enough to fit in an image block. This paper proposes an elegant solution that helps Wong's technique to be practically implemented. A new method of applying the cryptographic hash function is utilized. This method makes the image blocks able to hold longer and secure watermarks while providing similar level of the localization accuracy. The experimental results show that the proposed solution carries Wong's technique from a cryptographically vulnerable system to a secure and practical one.

1. INTRODUCTION

Digital watermarking is a technique for adding signals (watermark) to digital data (audio, video, or still images) that can be detected or extracted later to make an assertion about the data. If the data is manipulated, these watermarks will also be modified. This is because watermarks are embedded in the data content. An authenticator can examine the hidden watermark to verify the integrity of the data.

In 1998, Wong [1] proposed a public-key watermarking technique for image integrity verification. This technique was designed to detect and verify that the image has not been altered. In 2000, Wong and Memon [2] republished this technique with a minor modification that made the algorithm resist the Vector Quantization watermark attack [3]. One year later, the same authors in [4], recommended using an image block of size 12x12 in order to hold the full length of the hash function MD5 [5] output. Typically, these techniques utilized the public-key cryptosystem to sign the image data.

Public-key cryptosystems scramble or transform data to another form (domain) in a process called encryption. At the other end, this information may be recovered through a decryption process. Public-key cryptosystems use one key to encrypt the information and a different key to decrypt it.

One of the most well-known and popular public-key systems is called RSA [6]. The outline of RSA can be summarizes as follows:

- Two large prime numbers p and q are randomly generated and their product is calculated, denoted by $n = p \cdot q$
- A large integer d is randomly chosen, which is relatively prime to $(p-1)\cdot(q-1)$, i.e., the greatest common divisor of d and the product $(p-1)\cdot(q-1)$ is equal to 1 .
- The integer e is finally computed to satisfy that $(e \cdot d, \ \bmod \ (p-1)\cdot(q-1)) = 1$
- The public key is the pair (e, n) ; whereas the private key is the triple (d, p, q) .

The RSA encryption for any message is performed as follows:

- Represent the message as an integer $M \in [0, n-1]$.
- The encrypted message C is obtained by raising M to the power e modulo n , i.e.,

$$C = (M^e, \bmod n) \tag{1}$$

To decrypt the ciphertext C, raise it to the power d mod n , i.e.,

$$M = (C^d, \bmod n) \tag{2}$$

Note that, due to modulus arithmetic, the size of the ciphertext C always lies between 0 and $n-1$. Equations (1) and (2) are always true, as described in [1]. It is almost impossible to calculate d if only the public key (e, n) is known. Therefore to find d , the two primes p and q must be known. Since only n is publicly available, a cryptanalyst must determine p and q from n , which is called a factorization problem. Based on the contemporary computational methods and computer systems that were available when RSA appeared, a 100-digit length was long enough for n to be a hard factorization. At that time, Rivest challenged the world to factor RSA-129 [7]. He estimated that this would take about 10^{16} years of computing time. Note that, RSA-129 means that the length of n is 129 decimal digits or 429 bits). Seventeen years later, Atkins *et. al*. [8] showed that it took only eight months in a worldwide cooperative effort to do the job. In 1999, RSA-140 was factorized by Cavallar *et. al.* [9]. Recently a new record for the general factorization of RSA-155 (512 bits modulus) was achieved, again by Cavallar *et. al.* [10]. Hence the current recommendations from RSA specialists are that, decimal numbers of at least 232 digits (768-bits) should be used as keys to make RSA safe from factorization.

This paper provides a modification on Wong's technique to make it so secure that it would be practically implemented. The paper is organized as follows. In Section 2 a brief description of

0-7803-8554-3/04/$20.00 ©2004 IEEE.