TOWARD A SECURE PUBLIC-KEY BLOCKWISE FRAGILE AUTHENTICATION WATERMARKING

Paulo S. L. M. Barreto*, Hae Yong Kim*, Vincent Rijmen**

*Univ. São Paulo, Escola Politécnica, Av. Prof. Luciano Gualberto, tr. 3, 158, 05508-900, São Paulo, SP, Brazil.

**Cryptomathic, Lei 8a, B-3000 Leuven, Belgium.

E-mails: {paulob,hae}@lps.usp.br, VincentRijmen@cryptomathic.com

ABSTRACT

In this paper, we describe some weaknesses of public-key blockwise fragile authentication watermarkings and the means to make them secure. Wong's original algorithm as well as a number of its variant techniques are not secure against a mere block cut-and-paste or the well-known birthday attack. To make them secure, some schemes have been proposed to make the signature of each block depend on the contents of its neighbouring blocks. We attempt to maximise the change localisation resolution using only one dependency per block with a scheme we call hash block chaining version 1 (HBC1). We then show that HBC1, as well as any neighbour content-dependent scheme, are susceptible to another forgery technique that we have named a transplantation attack. We also show a new kind of birthday attack that can be effectively mounted against HBC1. To thwart these attacks, we propose using a nondeterministic digital signature together with a signature-dependent scheme (HBC2). Finally, we discuss the advantages of using discrete logarithm signatures instead of RSA for watermarking.

1. INTRODUCTION

A digital watermark is a visually imperceptible, information-carrying signal embedded in a digital image. A watermarking scheme can be classified as either *robust* or *fragile*. Robust watermarks are generally used for copyright and ownership verification. In comparison, fragile watermarks are useful for purposes of authentication and integrity attestation. A fragile watermark provides a guarantee that the image has not been tampered with and came from the right source. Many fragile watermarking schemes have been proposed in recent years, for example [1-5]. Among them, Wong in [2] has proposed a blockwise fragile authentication watermarking and in [3] has improved it by using a public-key based scheme. Since then, some number of public-key based fragile watermarks have

appeared in the literature [6-8]. Using a public-key cipher, claims of image authenticity can be judged without the necessity of disclosing any private information. Moreover, solid cryptography theory makes this scheme reliable, when due cares are taken into account. The present paper will discuss what these "due cares" are.

A digital signature [9, section 1.6] is an algorithm for ensuring integrity and authenticity of sensitive digital data. It computes a fingerprint of the data by using a hashing function, and then employs an asymmetric (public-key) cipher to encrypt the fingerprint with the originator's private-key. In the signature verification step, the hashing function is applied on the received data and the accompanying signature is decrypted using the signer's public-key. The results are expected to match, unless the data or signature are corrupted or faked.

Classical digital signatures are able to detect alterations in signed data but not to locate them. In contrast, most fragile watermarking techniques provide the ability to localise where the alterations have taken place, for this is obviously a desirable property. Wong proposed dividing an image into blocks and independently signing each block. The signature is then embedded in the least significant bit (LSB) of every pixel in the image. This scheme makes it possible to localise where the alterations are situated, but it presents many flaws. One of such flaws is its weakness against a mere block cut-and-paste attack (see figure 2) and Holliman and Memon's *counterfeiting attack* [10]. Even the recent technique proposed by Li et al. [6] fails to hinder these attacks. Another flaw of Wong and Li's techniques is the weakness against the well-known *birthday attack*.

Holliman and Memon [10] and, independently, ourselves [7-8] conclude that the use of contextual information can mend some of the weaknesses of blockwise-independent watermarking schemes. Using contextual information, the signature of a block is considered valid only if it is surrounded by correct blocks (see figure 1). In this case, if a block *B* is changed, the signature verification will fail in all those blocks that depend on *B*. Thus, a number as small as possible of dependencies is desirable for an accurate localisation of image changes. In the present paper, we propose making the signature of each block depend on only one other block, in order to maximise the change localisation resolution. The precise scan order does not matter. For example, zig-zag-scan (figure 1c) or raster-scan (figure 1d) are both well-suited. We call this scheme *hash block chaining*, version 1 (HBC1), reminiscent of the cipher block chaining construction [9, algorithm 7.13].

Nevertheless, Holliman and Memon [10] did not notice that any context-dependent scheme (including HBC1) is susceptible to another kind of forgery technique that we call a *transplantation*

attack. Moreover, although a classic birthday attack cannot be performed against HBC1, we will present a new improved birthday attack that can effectively be mounted against HBC1. We will show that an improved form of hash block chaining, HBC2, which makes use of nondeterministic digital signature (instead of RSA, which is deterministic) and signature-dependency (instead of content-dependency), can prevent these kinds of attack.

For ease of exposition, we will assume that the content being watermarked is a still image, though the techniques expounded are equally applicable to audio and video data. If the digital signatures are not to be embedded into the data (i.e., stored independently), the exposition also can be applied to any kind of data. Many of the ideas presented here have appeared in our earlier conference paper [11].

2. WONG'S SCHEME

Wong's public-key scheme [3] for watermark insertion in a greyscale image can be summarised as follows:

1. Let Z be an $N \times M$ image to be watermarked. Partition Z into n blocks Z_t ($0 \le t < n$) of 8×8 pixels (at most; border blocks may be shorter). Each Z_t will be watermarked separately.
2. Let A be a visually meaningful binary image to be used as watermark. This image is replicated periodically to get an image large enough to cover Z. To each block Z_t there will be a corresponding binary block A_t.
3. Let Z_t^* be the block obtained from Z_t by clearing the LSB of all pixels. Using a cryptographically secure hashing function H, compute the fingerprint $H_t \equiv H(M, N, Z_t^*)$.
4. Exclusive-or H_t with A_t, getting the marked fingerprint $\hat{H}_t$.
5. Encrypt $\hat{H}_t$ with the private key, thus generating a digital signature S_t.
6. Insert S_t into the LSB of Z_t^*, obtaining the marked block Z_t'.

The corresponding watermark verification algorithm is straightforward:

1. Let X' be an $N \times M$ watermarked image. Partition this image into n blocks X_t', as before.
2. Let X_t^* be the block obtained from X_t' by clearing the LSB of all pixels. Using the hashing function H chosen for insertion, compute the fingerprint $H_t \equiv H(M, N, X_t^*)$.

3. Extract the LSB from X'_t and decrypt the result using the public key, obtaining the decrypted block D_t.

4. Exclusive-or H_t with D_t, obtaining the check block C_t.

5. If C_t and A_t are equal, the watermark is verified. Otherwise, the marked image X' has been modified at block X'_t.

Here and throughout the remainder of this paper, the operator * indicates LSB clearing and the mark $'$ indicates a signature-inserted block or image. Notice that, theoretically, the image A must be publicly available for the verification to take place. In practice, however, A is a meaningful logo image and any change in X'_t will most likely generate a noise-like block C_t, that cannot be mixed up with A_t, even if A is not available (see figure 2). The image A may even be a completely black (or white) image, as suggested in [6], and in this case A can be easily made publicly available.

Li [6] suggests a slight variation of the scheme above. His method partitions each block into two halves. Then, the right half of a block Z_t^* is replaced with the right half of the next block $Z^*_{(t+1) \bmod n}$ along zig-zag-scan path (figura 1c) so that neighbouring blocks are related by blended data. Each combined block is then encrypted and embedded into the LSBs of the block Z_t^*. The same operations as those did on the encrypting side should be performed on the decrypting side.

3. SIMPLE ATTACKS AND COUNTERFEITING ATTACK

We now point out some cryptanalytical weaknesses of Wong's[1] and Li's methods and suggest the means to make them robust. An authentication scheme that succeeds to detect any change in the marked image should to be considered more secure than another that fails to detect some kinds of alterations, even if these alterations cannot be seemingly used for any malicious purposes. The mere existence of such flaws indicates a weakness in the scheme. They may be used in the future to attack the watermarking, even though by now no one knows how to do it.

For example, a greyscale watermarking technique is usually generalised to colour images by simply applying the method independently to the three colour planes (for example, [2-3]). In this case, the watermarking will not detect the swapping of the colour planes. Although it may be hard to

[1] We remark that 64-bit RSA, originally suggested for use with Wong's scheme, is completely insecure. RSA keys this size can be factored within seconds on a modern PC.

imagine how this attack could be used maliciously, it is more secure that even this sort of alteration should not pass undetected. This concrete problem can be easily overcome by hashing together the three colour planes.

There is another very simple attack, undetectable by Wong's watermarking scheme, that can really be used with malicious intentions. We have named it a *cut-and-paste attack*. Suppose an attacker has a collection of legitimately watermarked images, all of them of the same size and containing the same embedded image *A* in the watermark. Since each block is marked separately without any further information about the container image except its dimensions, it is possible for this attacker to select blocks from the authentic images and build with them a new image whose watermark will be falsely verified as legitimate. Here we assume that the original coordinates of each block are kept in the faked image. However, in some cases (for example, if the size of image *A* is 4×4, 4×8, 8×4, 8×8, 8×16, etc.) it might even be possible to cut-and-paste within a marked image while keeping the embedded watermark unchanged. Figure 2 shows an example of this attack.

This attack also applies to Li's watermarking: the attacker have only to copy LSB-cleared contents of two half-blocks from two neighbouring blocks, say X_t^* and X_{t+1}^*, and paste them together with the digital signature found in the LSBs of block X_t'.

If the cut-and-paste attack is repeatedly applied, a whole faked but validly-watermarked image can be constructed. This is the very idea of Holliman and Memon's *counterfeiting attack*. Let us suppose that an attacker wants to watermark an image *B* having in hand a database images protected by Wong's watermarking. The attacker first partitions *B* into blocks B_t. Let us suppose that watermark A_t is the logo that should be inserted into block B_t. The attacker searches, among database blocks containing watermark A_t, the block D_t' most similar to B_t. Then, the block D_t' is inserted in place of B_t. Repeating this process for all blocks of *B*, a faked (but correctly watermarked) image is constructed. This attack can succeed even using an astonishingly small database. Holliman and Memon took two 750×750 NIST greyscale fingerprint images, inserted the Wong's watermark in one of them and then constructed a convincing validly-watermarked approximation of the second image using the first as the database, that is, using only approximately 9000 database blocks. A similar attack can also be mounted against Li's watermarking.

We will show in section 5 that HBC1 makes impossible cut-and-paste and counterfeiting attacks.

4. SIMPLE BIRTHDAY ATTACK

Birthday attacks [9, section 9.7] constitute a well-known and powerful means of subverting digital signatures. The attacker searches for collisions, i.e. pairs of blocks that hash to the same value, thus having the same signature. Using a hashing function that produces *m* possible values, there is more than 50% chance of finding a collision whenever about $\sqrt{m}$ blocks are available. Wong's scheme uses a hashing function of no more than 64 bits; hence collisions are expected to be found when the attacker has collected merely about 2^{32} blocks. In general, the only protection against birthday attacks is to increase the hash size. This would decrease the change localisation resolution, because the blocks must be made larger to host more embedded data. We will show in the next section that a classical birthday attack also turns out to be impossible under HBC1.

A possible scenario for a birthday attack is an insurance company that keeps an incident image database using Wong's watermarking for image integrity and authenticity protection. A typical database of a large insurance company may contain over a million images with, say, 640×480 pixels, so that each image is partitioned into 4800 individually signed blocks (of 8×8 pixels). This results about 2^{32} signatures, enough for a birthday attack.

The attack proceeds as follows. An attacker wishing to replace a watermarked block X_t' by another block *B* prepares $r \approx 2^{32}$ visually equivalent variants $B_1, \ldots, B_r$ of *B*. This can be accomplished by varying the second least significant bit of each of 32 arbitrarily chosen pixels of *B* (the LSB cannot be used since the watermark will be stored there). The attacker then looks for an image block D' in the image database that hashes to the same value as any one of the B_j, i.e., such that

$$H(M, N, B_j^*) = H(M, N, D^*) .$$

The probability of success exceeds 0.5 because of the birthday paradox. This B_j (with the signature taken from D') can replace X_t' without being noticed by Wong's scheme. If this process is repeated a sufficient number of times, a whole faked image can be created. Similar attack can also be mounted against Li's watermarking.

5. HASH BLOCK CHAINING VERSION 1

As pointed out in [7-8; 10], the solution to hinder many simple attacks is to introduce contextual information. That is, in the computation of the fingerprint H_t, feed the hashing function *H* with the

neighbouring blocks of Z_t^*, besides the block Z_t^* itself (see figure 1). In this case, if a block X_t' is altered, signature verification will fail in all those blocks that depend on X_t', besides in block X_t' itself. Thus, a number as small as possible of dependencies is desirable for an accurate localisation of image changes; ideally, a single dependency per block. The following scheme implements this idea:

$$H_t \equiv H(M, N, Z_t^*, Z_{(t-1) \bmod n}^*, t) .$$

The block index *t* was inserted in order to detect blockwise rotation. As in Wong's scheme, image sizes *M* and *N* are inserted to detect image cropping. We call this construction *hash block chaining*, version 1 (HBC1). We stress that if a block X_t' is altered, then HBC1 will report that $X'_{(t+1) \bmod n}$ is invalid (besides X_t' itself).

Using HBC1, the simple cut-and-paste attack can no more be perpetrated, because if a spurious block is pasted in place of X_t', with very high probability this alteration will introduce a change in $H_{(t+1) \bmod n}$. The probability of such a change not taking place is only $O(m^{-1})$. This change invalidates the signature of the block $X'_{(t+1) \bmod n}$. Thus, the cut-and-paste attack (and consequently, the counterfeiting attack) can no more be perpetrated.

Similarly, if a birthday attack is performed, the changed contents of X_t' induce with high probability a change in $H_{(t+1) \bmod n}$. Thus, the attacker will have to forge the signature of $X'_{(t+1) \bmod n}$ as well, perpetrating another attack. But this induces a change in $X'_{(t+2) \bmod n}$. Therefore, the attacker will face the problem that bad signatures propagate cyclically over all blocks, eventually destroying the forged signature of the very first faked block.

6. TRANSPLANTATION ATTACK

HBC1 is effective against cut-and-paste, counterfeiting and simple birthday attacks. But it is not secure against an improved form of cut-and-paste attack described below. Indeed, HBC1 or *any* other partitioning technique that augments the hashing function input with deterministic, limited context from the neighbouring blocks are susceptible to what we call a *transplantation attack*. To see why this holds, let X' and $\overline{X}'$ be two HBC1-watermarked images. Let $X'_A \rightarrow X'_B$ denote the fact that the hashing of block X'_B depends on the contents of block X'_A (that is, on X_A^*). Suppose that images X' and $\overline{X}'$ have blocks as shown below:

$$\cdots \to X'_A \to X'_D \to X'_B \to X'_C \to \cdots,$$

$$\cdots \to \overline{X}'_A \to \overline{X}'_E \to \overline{X}'_B \to \overline{X}'_C \to \cdots,$$

where $X^*_A = \overline{X}^*_A$, $X^*_B = \overline{X}^*_B$, $X^*_C = \overline{X}^*_C$ but $X^*_D \neq \overline{X}^*_E$. Then the pair of blocks (X'_D, X'_B) can be interchanged with pair $(\overline{X}'_E, \overline{X}'_B)$, without being detected by the HBC1 scheme:

$$\cdots \to X'_A \to \overline{X}'_E \to \overline{X}'_B \to X'_C \to \cdots,$$

$$\cdots \to \overline{X}'_A \to X'_D \to X'_B \to \overline{X}'_C \to \cdots.$$

Document images usually have large white areas, which makes them very susceptible to transplantation attacks. For example, if X'_A, X'_B, X'_C, $\overline{X}'_A$, $\overline{X}'_B$ and $\overline{X}'_C$ were all completely white noiseless blocks, the assault would easily succeed. Note that merely increasing the number of dependencies does not prevent the transplantation attack. If there were two dependencies per block, as illustrated below, the triple of blocks (X'_B, X'_E, X'_C) would be interchangeable with the triple $(\overline{X}'_B, \overline{X}'_F, \overline{X}'_C)$.

$$\cdots \leftrightarrow X'_A \leftrightarrow X'_B \leftrightarrow X'_E \leftrightarrow X'_C \leftrightarrow X'_D \leftrightarrow \cdots,$$

$$\cdots \leftrightarrow \overline{X}'_A \leftrightarrow \overline{X}'_B \leftrightarrow \overline{X}'_F \leftrightarrow \overline{X}'_C \leftrightarrow \overline{X}'_D \leftrightarrow \cdots.$$

Similar attacks can be performed against 4 dependencies or 8 dependencies per block as well.

7. IMPROVED BIRTHDAY ATTACK

HBC1 cannot withstand a more sophisticated birthday attack either. This attack replaces simultaneously two consecutive blocks X'_t and X'_{t+1} by forged blocks B_t and B_{t+1} (we will omit "mod n" in the indices to simplify the notation.) Three fingerprints are affected by these substitutions: H_t (which depends on X'_t), H_{t+1} (which depends on both X'_t and X'_{t+1}), and H_{t+2} (which depends on X'_{t+1}). Suppose that the database has s signed blocks.

The attacker prepares p visually equivalent variants for B_t. Then, likely $P \cong ps/m$ collisions for H_t will be found (see [12]). More explicitly, P pairs (B^1_t, D^1_t), ..., (B^P_t, D^P_t) will be found, where B^1_t, ..., B^P_t are visually equivalent variants of B_t and D^1_t, ..., D^P_t are database blocks such that the fingerprint of D^i_t is the same as the fingerprint of B^i_t. That is:

$$H(M, N, D^{i*}_t, X^*_{t-1}, t) = H(M, N, B^{i*}_t, X^*_{t-1}, t), \text{ for } 1 \le i \le P.$$

Consequently, the signature of block t will stay valid if X_t is replaced by any block B_t^{i*} together with the signature taken from LSBs of D_t^i. Nevertheless, almost certainly this replacemente will make invalid the signature of block t+1.

Similarly, the attacker prepares q variants for B_{t+1}, likely yielding $Q \cong qs/m$ collisions for H_{t+2}. Let (B_{t+1}^1, D_{t+2}^1), ..., (B_{t+1}^Q, D_{t+2}^Q) be these colliding pairs, that is:

$$H(M, N, X_{t+2}^*, B_{t+1}^{j*}, t+2) = H(M, N, X_{t+2}^*, D_{t+2}^{j*}, t+2), \text{ for } 1 \le j \le Q.$$

The signature of block t+2 will stay valid if X_{t+1} is replaced by any B_{t+1}^{j*} together with the signature taken from LSBs of D_{t+2}^j. But this replacement will probably make invalid the signature of block t+1.

Combining all colliding variants of B_t and B_{t+1} will yield about $(ps/m)(qs/m) = pqs^2/m^2$ pairs (B_t^i, B_{t+1}^j), visually equivalent to (B_t, B_{t+1}). Now, the attacker has to find a collision for H_{t+1}, that is, has to find one variant pair (B_t^i, B_{t+1}^j) and one database block D_{t+1} such that:

$$H(M, N, B_{t+1}^{j*}, B_t^{i*}, t+1) = H(M, N, D_{t+1}^*, B_t^{i*}, t+1).$$

Then, if X_t and X_{t+1} are replaced by forged blocks B_t^{i*} and B_{t+1}^{j*} and, at the same time, the signatures of blocks t, t+1 and t+2 are replaced by the signatures taken from LSBs of D_t^i, D_{t+1} and D_{t+2}^j, the forgery will pass undetected by HBC1.

How large shall p and q be to make the chance of success exceed 50%? As there are pqs^2/m^2 pairs of blocks and s database blocks, a collision for H_{t+1} will likely occur when $(pqs^2/m^2)s \approx m$, i.e. when $pq \approx (m/s)^3$. Thus, if the database has $s \approx \sqrt{m}$ valid signatures, probably two faked blocks can replace two valid consecutive blocks when $p \approx q \approx m^{3/4}$ visually equivalent variants of each faked block are prepared.

8. HASH BLOCK CHAINING VERSION 2

We have improved HBC1 to thwart both transplantation and improved birthday attacks. This enhanced version was named HBC2 and it makes use of *nondeterministic* signature schemes. Some signature schemes (for example, DSA and Schnorr's scheme [9, section 11.5]) are nondeterministic in the sense that each individual signature depends not only on the hashing function, but also on some randomly chosen parameter. Using a nondeterministic signature algorithm, even the signatures

of two identical images will be different. This property effectively prevents transplantation attacks. A deterministic signature (like RSA) can be converted into a nondeterministic one by appending "salt" (i.e., arbitrary, statistically unique data) to the message being signed. HBC2 is defined as follows:

$$H_t \equiv H(M, N, Z_t^*, Z_{(t-1) \bmod n}^*, t, S_{t-1}),$$

where S_{t-1} is the nondeterministic signature of block Z_{t-1}, and $S_{-1} \equiv \varnothing$. Note that we cannot use $S_{(t-1) \bmod n}$ because by the time H_0 is being computed, S_{n-1} would not be known yet.

The improved birthday attack is completely ineffective against HBC2, because in HBC2 the signature of one block depends not only on the content of its neighbouring block, but also on its nondeterministic signature. Let us suppose that an attacker has managed to replaced two valid consecutive blocks X_t and X_{t+1} by two faked blocks B_t and B_{t+1}, and three signatures S_t, S_{t+1} and S_{t+2} by three faked (but valid) signatures L_t, L_{t+1}, L_{t+2} while maintaining intact the content of the block X_{t+2}. Note that this replacement is much harder for HBC2 than for HBC1 due to the nondeterministic signature and the signature-dependency. Even in this improbable scenario, HBC2 will report an alteration, because H_{t+3} depends not only on the content of $X_{t+2,}$ which is left untouched, but also on its signature, which almost certainly changes.

The use of HBC2 has a surprising pleasant side effect. Typically, birthday attacks can be mounted against hashing functions of length m with an effort of $O(\sqrt{m})$ steps. However, for HBC2 no attack that takes less than $O(m)$ steps is known. Therefore it seems that, in an optimistic scenario, the hash length could be cut in half while keeping the original security level. However, we don't recommend reducing the hash length until this conjecture is scrutinised in greater depth, as such a reduction might adversely affect the security of the signature algorithm itself.

HBC2 is capable of detecting whether any blocks have been changed, rearranged, deleted, inserted, or transplanted from a legitimately signed image. Besides, it either indicates which blocks were altered or, if a large validly watermarked region is copied and pasted, where the borders of this region lie. We notice that the location capability is lost if a block (or a row or a column) is inserted or deleted, though even in this case HBC2 will correctly report the presence of some alteration.

9. DISCUSSIONS AND EXPERIENCES

Typically, the length of a discrete logarithm signature is about twice the length of the hash used [9, section 11.5]. This is better than RSA signatures, whose length is always that of the public key. For

instance, DSA signatures are 320 bits in length, while RSA signatures with equivalent security level must be about 1024 bits long. In this sense, Schnorr signatures are best suited for HBC2 [9, section 11.5.3], as they achieve maximum reduction in signature size and hence in the amount of data to be embedded in the host image.

Experiences with HBC2 using elliptic curve cryptography yielded signing and verifying times of about 10 seconds on a Pentium-500, for 512×512 greyscale images. The change location uncertainty was smaller than 0.2% of the image area.

10. CONCLUSION

In this paper, we advanced some more steps toward a really secure blockwise fragile authentication watermarking. We took Wong and Li's algorithms and showed them to be insecure against attacks as simple as block cut-and-paste and the well-known birthday attack. We proposed the HBC1 scheme, which counters these attacks by making the signature of each block depend on the contents of a neighbouring block. Then we showed how HBC1, as well as any scheme that augments the hashing input with the contents of neighbouring blocks, is susceptible to the transplantation attack. We also presented a new improved birthday attack that does apply to HBC1. To thwart these attacks, we defined HBC2 using nondeterministic signature and signature-dependency, and argued its effectiveness against transplantation and improved birthday attacks. Finally, we discussed the advantages of using discrete logarithm signatures and presented some experimental data.

11. ACKNOWLEDGEMENTS

Two of authors, Barreto and Kim, would like to thank FAPESP for the partial financial support of this work under grant 2001/02400-9.

12. REFERENCES

[1] Yeung, M.M., and Mintzer, F.: 'An Invisible Watermarking Technique for Image Verification'. Proceedings of *IEEE International Conference on Image Processing*, 1997, vol. 1, pp. 680-683.

[2] Wong, P.W.: 'A Watermark for Image Integrity and Ownership Verification'. Proceedings of *IS&T PIC Conference*, (Portland, OR), May 1998 (also available as Hewlett-Packard Labs. Tech. Rep. HPL-97-72, May 1997).

[3] Wong, P.W.: ‘A Public Key Watermark for Image Verification and Authentication’. Proceedings of *IEEE International Conference on Image Processing*, 1998, vol. 1, pp. 455-459, (MA11.07).

[4] Bhattacharjee, S., and Kutter, M.: ‘Compression Tolerant Image Authentication’. Proceedings of *IEEE International Conference on Image Processing*, 1998, vol. 1, pp. 435-439, (MA11.03).

[5] Wu, M., and Liu, B.: ‘Watermarking for Image Authentication’. Proceedings of *IEEE International Conference on Image Processing*, 1998, vol. 2, pp. 437-441, (TA10.11).

[6] Li, C.T., Lou, D.C., and Chen, T.H.: ‘Image Authentication and Integrity Verification via Content-Based Watermarks and a Public Key Cryptosystem’. Proceedings of *IEEE International Conference on Image Processing*, 2000, vol. 3, pp. 694-697, (WP06.10).

[7] Barreto, P.S.L.M., and Kim, H.Y.: ‘Pitfalls in Public Key Watermarking’. Proceedings of *Sibgrapi - Brazilian Symposium on Computer Graphics and Image Processing*, 1999, pp. 241-242.

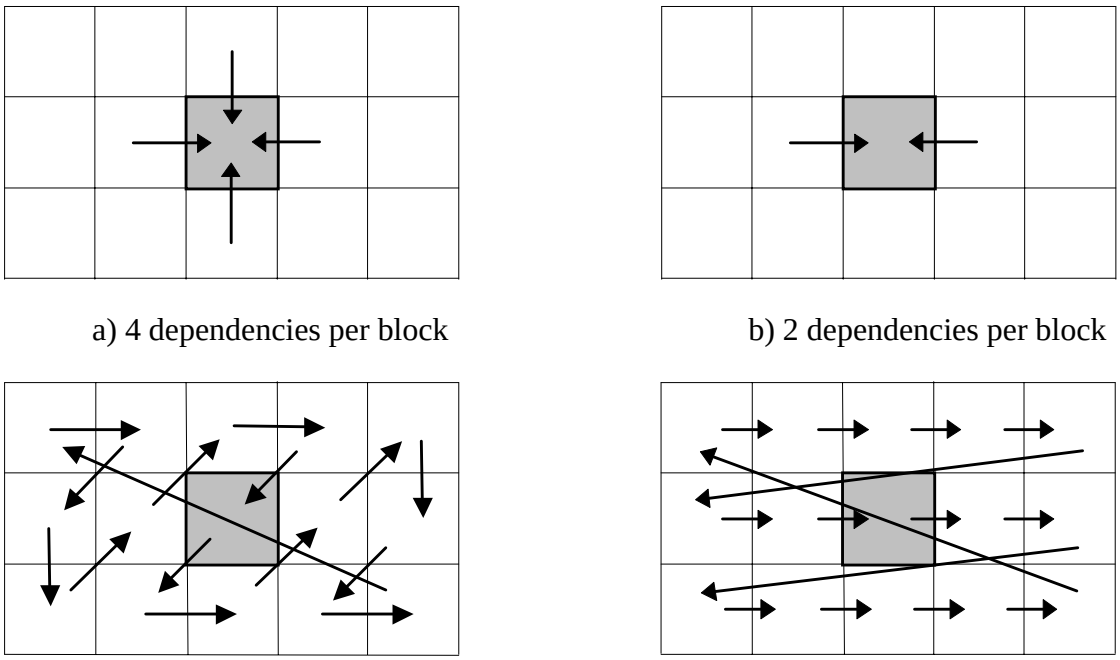
[8] Barreto, P.S.L.M., Kim, H.Y., and Rijmen, V.: ‘Um Modo de Operação de Funções de Hashing para Localizar Alterações em Dados Digitalmente Assinados’. Proceedings of *Simpósio Brasileiro de Telecomunicações*, 2000, paper #5150124.

[9] Menezes, A.J., Van Oorschot, P.C., and Vanstone, S.A.: ‘Handbook of Applied Cryptography’ (CRC Press, 1997)

[10] Holliman, M., and Memon, N.: ‘Counterfeiting Attacks on Oblivious Block-wise Independent Invisible Watermarking Schemes’. *IEEE Trans. Image Processing*, 2000, vol. 9. no. 3, pp. 432-441.

[11] Barreto, P.S.L.M., Kim, H.Y., and Rijmen, V.: ‘Toward a Secure Public-Key Blockwise Fragile Authentication Watermarking’. To appear in Proceedings of *IEEE International Conference on Image Processing*, 2001.

[12] Nishimura, K. and Sibuya, M.: ‘Probability to Meet in the Middle’. *Journal of Cryptology*, 1990, vol. 2, no. 1, pp. 13-22.



a) 4 dependencies per block

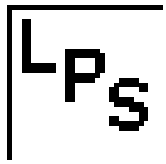
b) 2 dependencies per block

c) 1 dependency per block (zig-zag-scan)

d) 1 dependency per block (raster-scan)

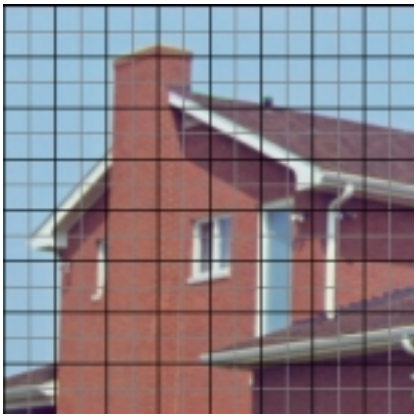
Fig. 1: The use of contextual information. To compute the signature of a block *B* (shown in grey), the contents of *B* and its neighbouring blocks are taken into account. HBC uses 1 dependency per block, either zig-zag or raster scan.

a) Original image



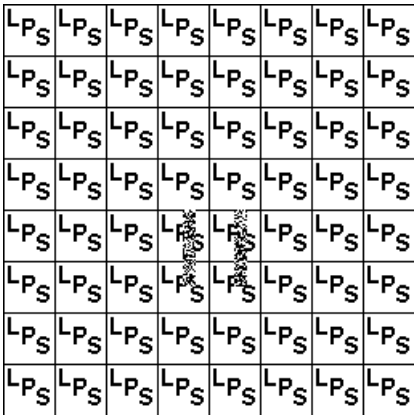
b) 32×32 logo image

c) HBC2-watermarked image



d) 16×16 and 32×32 blocks

e) Cut-and-paste attack



f) Delimiting the alteration

Fig. 2: Hindering cut-and-paste attack with HBC2. A 256×256 original colour image (a) was marked using the private key and a 32×32 logo image (b), yielding watermarked image (c). The image (d) shows its constituent blocks. The watermarked image (c) suffered a cut-and-paste attack (e), undetectable by Wong's scheme. Using HBC2, the altered blocks can be located (f). Notice that HBC2 detects only borders of changed 16×16 blocks.