TOWARD A SECURE PUBLIC-KEY BLOCKWISE FRAGILE AUTHENTICATION WATERMARKING

Paulo S. L. M. Barreto*, Hae Yong Kim*, Vincent Rijmen**
**Univ. São Paulo, Dept. Eng. Sist. Eletrônicos, Av. Prof. Luciano Gualberto, tr-3, 158, 05508-900, São Paulo, Brazil.*
***FWO postdoctoral researcher, sponsored by the National Fund for Scientific Research — Flanders (Belgium);*
Katholieke Universiteit Leuven, Dept. Elektrotechniek-ESAT, Kasteelpark Arenberg, 10, Heverlee, Belgium.
E-mail: {paulob,hae}@lps.usp.br, vincent.rijmen@esat.kuleuven.ac.be

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 is not secure against a mere block cut-and-paste or the well-known birthday attack. To make it secure, some schemes have been proposed to make the signature of each block depend on the contents of its neighboring blocks. We attempt to maximize the change localization 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* neighbor-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, for example [1-3]. Among them, Wong has proposed using a public-key based digital signature scheme [1-2]. 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 [4, 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.

The ability to localize where the alterations have taken place is obviously a desirable property. Classical digital signatures are able to detect alterations in signed data but not to locate them. 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 localize where the alterations are situated, but it presents many flaws. One of such flaws is its weakness against the "birthday attack," as pointed out by Holliman and Memon [5] and, independently, by ourselves [6-7]. Wong's scheme is also insecure against a mere block cut-and-paste attack (see figure 2).

The works [5-7] 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 localization of image changes. In the present paper, we propose making the signature of each block depend on only one other block, in order to maximize the change localization resolution. We call this scheme *hash block chaining*, version 1 (HBC1), reminiscent of the cipher block chaining construction [4, algorithm 7.13].

Holliman and Memon [5] 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 and signature-dependency, can prevent these kinds of attack.

2. WONG'S SCHEME

Wong's scheme for watermark insertion in a grayscale image can be summarized 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 k, thus generating a digital signature $S_t = E_k(\hat{H}_t)$.

6. Insert S_t into the LSB of Z_t^*, obtaining the marked block X_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.

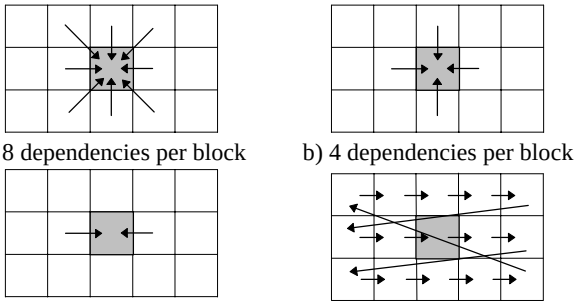
2. Let X_t^* be the block obtained from X_t by clearing the LSB of all pixels. Using the hashing function H chosen for insertion, compute the fingerprint $H_t \equiv H(M, N, X_t^*)$.

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.

Notice that, theoretically, the image A must be publicly available for the verification to take place. In practice, however, A is a meaningful 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).



a) 8 dependencies per block b) 4 dependencies per block

c) 2 dependencies per block d) 1 dependency per block

Fig. 1: The use of contextual information. To compute the signature of a block B (shown in gray), the contents of B and its neighboring blocks are taken into account. Figure (d) shows the chain of dependencies in HBC.

3. SIMPLE ATTACKS

We now point out some cryptanalytical weaknesses of Wong's method and suggest the means to make it robust[1]. Notice that an authentication scheme is really secure only if *any* change in the marked image is detectable, even if these changes 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 grayscale watermarking technique is usually generalized to color images by simply applying the method independently to the three color planes (for example, [1-2]). In this case, the watermarking will not detect the swapping of the color planes. Although it may be hard to 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 color planes.

[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.

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.

4. SIMPLE BIRTHDAY ATTACK

Birthday attacks [4, section 9.7] constitute a more sophisticated 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 localization resolution, because the blocks must be made larger to host more embedded data. We will show in the next section that HBC1 makes a classical birthday attack impossible.

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 Wong's watermark will be stored there). The attacker then looks for an image block C in the image database that hashes to the same value as any one of the B_j, i.e., such that

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

The operator * indicates LSB clearing. The probability of success exceeds 0.5 because of the birthday paradox. This B_j (with the watermark taken from C) 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.

5. HASH BLOCK CHAINING VERSION 1

As pointed out in [5-7], the solution to hinder cut-and-paste and birthday attacks is to introduce contextual information. That is, in the computation of the fingerprint H_t, feed the hashing function H with the neighboring 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 localization 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. 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 $X_{(t+1) \bmod n}$. 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 birthday attack. But this induces a change in $H_{(t+2) \bmod n}$. 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 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 neighboring blocks are susceptible to what we call a *transplantation attack*. To see why this holds, let X' and X'' be two HBC1-watermarked images. Let $X_A \to X_B$ denote the fact that the hashing of block X_B depends on the contents of block X_A. Suppose that images X' and X'' have blocks as shown below:

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

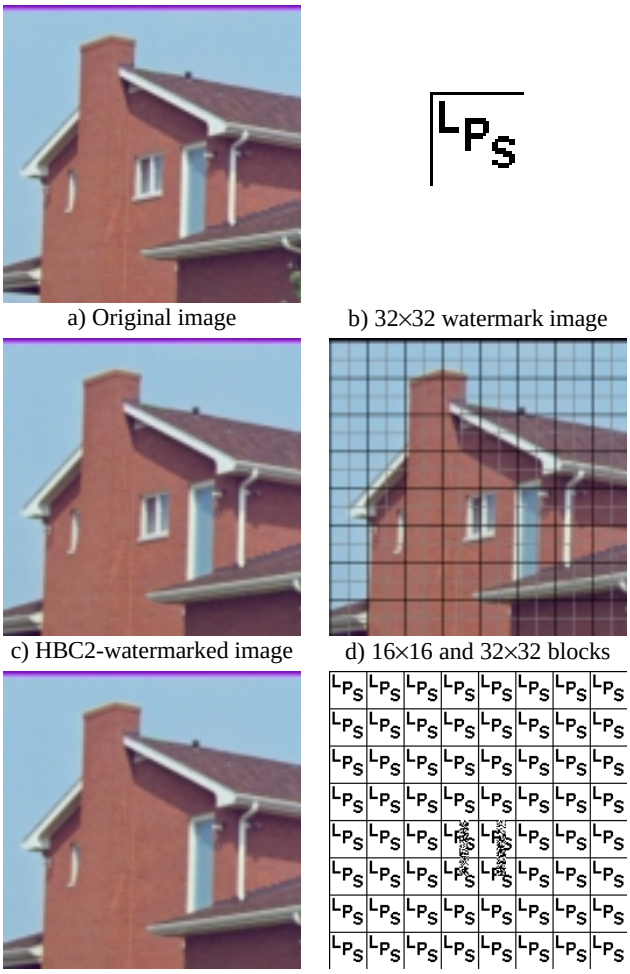
where the block contents of X'_A are identical to those of X''_A, the same holding for X'_B and X''_B, and for X'_C and X''_C, but *not* for X'_D and X''_E. Then the pair of blocks (X'_D, X'_B) can be interchanged with pair (X''_E, X''_B), without being detected by the HBC1 scheme:

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

Document images usually have large white areas, which makes them very susceptible to transplantation attacks. For example, if X'_A, X'_B, X'_C, X''_A, X''_B and 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 (X''_B, X''_F, X''_C).

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

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



a) Original image b) 32×32 watermark 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 color 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.

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 and q variants for B_{t+1}. Then, likely ps/m collisions for H_t and qs/m collisions for H_{t+2} will be found (see [8]). Combining all colliding variants of B_t and B_{t+1} will yield about $(ps/m)(qs/m)$ pairs visually equivalent to $(B_t,$

B_{t+1}). One of those variant pairs is likely to produce a collision for H_{t+1} (besides H_t and H_{t+2}) when $pqs^3/m^3 \approx 1$, 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 [4, 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 neighboring 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 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 scrutinized 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 their contents are valid, where the borders of the valid regions lie. We notice that the location capability is lost if a block 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 [4, 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 [4, 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 grayscale 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's algorithm and showed it 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 neighboring block. Then we showed how HBC1, as well as any scheme that augments the hashing input with the contents of neighboring 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. REFERENCES

[1] P. W. Wong, "A Public Key Watermark for Image Verification and Authentication," in *Proc. IEEE Int. Conf. Image Processing*, vol. 1, MA11.07, 1998.

[2] P. W. Wong, "A Watermark for Image Integrity and Ownership Verification," in *Proc. IS&T PIC Conf.*, (Portland, OR), May 1998 (also available as Hewlett-Packard Labs. Tech. Rep. HPL-97-72, May 1997).

[3] M. M. Yeung and F. Mintzer, "An Invisible Watermarking Technique for Image Verification," in *Proc. Int. Conf. Image Processing*, vol. 1, pp. 680-683, 1997.

[4] A. J. Menezes, P. C. Van Oorschot, S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.

[5] M. Holliman and N. Memon, "Counterfeiting Attacks on Oblivious Block-wise Independent Invisible Watermarking Schemes," *IEEE T. Image Processing*, vol. 9. no. 3, pp. 432-441, March 2000.

[6] P. S. L. M. Barreto, H. Y. Kim and V. Rijmen, "Um Modo de Operação de Funções de Hashing para Localizar Alterações em Dados Digitalmente Assinados," in *Proc. Simpósio Brasileiro de Telecomunicações*, paper #5150124, 2000.

[7] P. S. L. M. Barreto and H. Y. Kim, "Pitfalls in Public Key Watermarking," in *Proc. Sibgrapi - Brazilian Symp. on Comp. Graph. and Image Proc.*, pp. 241-242, 1999.

[8] K. Nishimura and M. Sibuya, "Probability to Meet in the Middle," *J. Cryptology*, vol. 2, no. 1, pp. 13-22, 1990.