Strict Integrity Control of Biomedical Images

Gouenou Coatrieux, Henri Maître and Bülent Sankur

ENST-Département TSI, 46 rue Barrault, 75013 Paris, France
LTSI-INSERM, Université de Rennes I, Campus de Beaulieu, 35042 Rennes Cedex, France
Boğaziçi University, Bebek, İstanbul, Turkey

ABSTRACT

The control of the integrity and authentication of medical images is becoming ever more important within the Medical Information Systems (MIS). The intra- and interhospital exchange of images, such as in the PACS (Picture Archiving and Communication Systems), and the ease of copying, manipulation and distribution of images have brought forth the security aspects. In this paper we focus on the role of watermarking for MIS security and address the problem of integrity control of medical images. We discuss alternative schemes to extract verification signatures and compare their tamper detection performance.

Keywords: Medical imaging, Watermarking, Strict integrity control, Hash functions, Hamming codes.

1. INTRODUCTION

The information infrastructure of modern health care is based on digital information management. While the recent advances in information and communication technologies provide new means to access, handle and move medical images, they also compromise their security due to their ease of manipulation and replication.1 The main aspects of medical data security with their concomitant risks are shown in Fig.1. In this context we focus on the security aspects of medical images and in fact we address the specific issue of the verification of their integrity. To this effect we use signatures extracted from the image coupled with watermarking techniques to insert them in order to realize the control of integrity.

The studies that are specifically directed to watermarking of medical images are few. Anand et al2 have proposed to embed an encrypted version of the Electronic Patient Record (EPR) in the least significant bit (LSB) plane of the image. While this scheme may seem to affect minimally the diagnostic content the ease with which the LSB plane can be manipulated is well known. Maou et al3 have similarly proposed a LSB technique where the host image authenticates the transmission origin with an embedded message composed of various patient data (e.g., ECG record), the diagnosis report and the doctor's seal. Macq and Dewey,4 propose a trusted header scheme by embedding the hash of the file header of medical standard image in the image raw data.

We propose an alternate approach to image integrity verification using watermarking where the protection zone is separated from the insertion zone to avoid compromising any diagnostic capability. Thus integrity of the original data is preserved in the diagnostic zone. Underlying in this approach is that medical images allow a separation of a ROI (Region of Interest) and a RONI (Region of Non-Interest) whose integrity needs not be preserved and serves as the watermark carrier. We explore three signature extraction alternatives and analyze their potential in precluding tampered and fabricated images which can escape detection. Watermarking techniques are being employed in increasing number of cases where document security integrity must be maintained,5-9 but as they are dedicated to be applied in E-commerce applications the image quality preservation issue is different than the medical one. It is thus conjectured that for hospital intra- and internets fragile watermarking will also provide the most appropriate toolbox to meet their security control challenges.

The structure of the paper is as follows: Section 2 contains a discussion on the relevance of watermarking in satisfying integrity verification of medical images. In Section 3 we look into the organization of image data structure, and general principles of control functions. The control function alternatives and their statistical performances are dealt with, respectively, in Sections 4 and 5. Conclusions are drawn in Section 6.

Corresponding author, G. Coatrieux e-mail: coatrieux@tsi.enst.fr
2. THE ROLE OF WATERMARKING FOR MEDICAL IMAGES

The ethics and legislative rules of medical deontology impose to the handling of medical data the following requirements (Fig.1):

- Confidentiality: The requirement that only the entitled users access to the information;
- Reliability to be understood as: i) Integrity, the verification that the information has not been modified by any non-authorized person, ii) Authentication: the proof that the information belongs to the claimed patient and comes from the correct source;
- Availability: ability of a MIS (Medical Information System) to be used in the normal scheduled conditions of access and exercise.

In this work we address only the problem of the strict integrity control of medical images. It is critical to determine the point at which the strict integrity control commences, in other words the stage in the processing chain at which the image is "frozen", beyond which its integrity is to be guaranteed. This point can be at the sensor stage or at the end of the preprocessing stage at which the radiologist esteems that the image quality is adequately enhanced for diagnostic purposes, i.e., post diagnosis and standardized image. From this point on, in the chain of operations, the watermarking techniques provide the right tools for protecting the image and verifying its integrity in the aftermath.

The security risks of medical images can vary from random errors occurring during transmission to lost or overwritten segments in the network during exchanges in the intra- and interhospital networks. One must also guarantee that the header of the image file always matches that of the image data. In addition to these unintentional modifications one can envision various malevolent manipulations to replace or modify parts of the image, called tampering. It is obvious that the integrity control of images can be addressed at two levels, that is, strict integrity control whereby one has to guarantee that the whole image is preserved as entire bit planes, or content-based control in which the pixels are allowed to differ to the extent that the visual content remains preserved.

In order to put the role of watermarking in the correct perspective we assume that all the other necessary security measures such as access control, encryption, auditing etc. are being exercised in the network. The need for watermarking comes in for the eventuality when all other security measures have been bypassed for the "clear" images circulating and being accessed in a network. In this respect watermarking provides the ultimate protection as it is inextricably attached to the document.

Finally let’s note that a byproduct of watermarking is the data hiding possibility, whereby metadata can be inserted into the image. Such supplementary information is intended to enrich or to facilitate the use of that image, for example it may consist of diagnostic annotations.

3. STRICT INTEGRITY CONTROL OF MEDICAL IMAGES

3.1. The Insertion Principle

We assume that the images to be protected can be separated into two zones: A zone, called ROI (Region of Interest), is the part that is utilized for the diagnosis whose integrity must be strictly controlled, where the modification of even one bit is not tolerated. The second zone, called the insertion zone or region of non-interest (RONI), is the peripheral region (outside the ROI) which does not contribute to the diagnosis. Figure 2 illustrates the case of
the two zones, the ROI and the insertion zones, on a brain MRI image. The ROI should be defined automatically or delineated semi-automatically with minimal interaction and in a standard way by the radiologist and the ROI boundary information should be available in the verification process. The assumption that images can invariably be separated into the so-called ROI and RONI zones does not hold for images in general, but for medical images it is a viable assumption.

The strict integrity control is achieved by extracting a signature from the ROI, which is then to be embedded in the insertion zone. Any image for which the signature extracted from the protected zone does not match with that stored in the insertion zone is declared invalid. Thus any change, whether due to malicious forging, tampering or to casual processing, random errors, etc. will give an alarm of integrity loss.

As the insertion zone does not contribute at all to the diagnosis, we have some liberty in handling this non-information bearing zone. Thus the ROI signatures can be inserted with adequate strength and/or necessary capacity,\textsuperscript{11} even beyond some visibility threshold, provided it does not become annoying or distracting to the radiologist. We assume that the insertion region can accommodate typically a few hundred bits. Obviously to foil any forgery attempt the signature could be encrypted and its insertion sites should be made secret based on some address selection key. At this stage the specification of the watermark insertion algorithm is not very critical since most existing algorithms would serve the purpose as the robustness and invisibility requirements are not very demanding. In summary the proposed scheme enables one to preserve intact the diagnostic zone while providing an insertion region with adequate capacity.

3.2. Signature Extraction

The image ROI can be considered as a bit volume and alternatively, depending on how it is scanned, as a binary sequence. Thus it becomes convenient to view the data to be protected as a binary message $M$, $|M|$ bits long, whose integrity is to be verified with an $|H|$ bit long binary signature $H$. The signature extraction function, $f$, is called in the sequel as the control function

$$ H = f(M). $$

Obviously $M$ could be subdivided in $N$ segments $m_i$, of $\mu_i$ bits each: $M = \{m_1, \ldots, m_N\}$. To authorize alteration localization, the signature $H$ results from the concatenation of the $N$ control words $h_i$ of lengths $\eta_i, \ldots, \eta_N$, each $h_i$ protects one segment $m_i$ independently from the others:

$$ H = f(M) = f((m_1, \ldots, m_N)) = (h_1, \ldots, h_N). $$

Without loss of generality we will assume that all the segments have the same size: $\mu = |M|/N$ and furthermore that the control functions $(f_1, \ldots, f_N)$ are all identical (i.e. $\eta_i = \eta_j$) as illustrated in 3.

The integrity is verified by comparing the embedded signature $H$ and the one calculated from the available image $H' = f(M')$. Obviously the control can be done at a global level whereby the whole image is invalidated if any one of $(h_1, \ldots, h_N)$ or more differs, that is $H \neq H'$, or by parts where only the segment $\{m_i\}$ is said to be tampered if $h_i \neq h'_i$. 

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{Figure2.png}
\caption{An MRI image where the protection zone (ROI) is delimited by an elliptic area.}
\end{figure}
Figure 3. Partitioning of the ROI message sequence \( M \) into \( N \) segments and the piecemeal generation of the control sequence \( H \), the signature to be embedded in the insertion zone.

The segments \( \{ m_k \} \) of \( M \) (segments of the message in the bit organization) can be obtained in many different ways. They may also carry information about different image features. We have selected three different features of interest:

1. Spatial localization
Segments resulting from concentric elliptical rings and angular sectors are illustrated in Fig. 4. They are well adapted for many applications like neurologic or pneumologic exams. Different shapes may be better adapted in other cases. It may be desirable that a manipulation on any part of the image have repercussions on several control words. This dispersion effect can be obtained by having the footprint of every segment \( \{ m_j \} \) not localized, but extending over several regions of the ROI. In fact in the extreme case one can constitute the segments \( \{ m_j \} \) by random permutations or scanning of the image pixels such that footprints are laid out randomly and uniformly over the whole ROI. Therefore manipulation of any part of the image should cause tamper alarm in all the control words. Alternatively two or more segmentation maps can be overlaid to refine the localization of the manipulated region as illustrated in Fig. 4. This, not only improves the detection performance but also reduces the substitution attack\(^{12}\) or collage attack leeway.

2. Grey level bit plane
In complement, the different bit planes of the image can be checked to differentiate LSB from MSB or middle bit planes modifications leading to the evaluation of the alteration depth.

3. Spatial extension
With this control we want to know if the attempt to modify the image is a high frequency modification of pixels or if it is distributed over a larger connected set of pixels. This may be well controlled by selecting an adequate subband in a frequential representation of the image.

By using segments originated from these three different approaches we may provide a diagnosis giving the spatial localization of the attack, its extension and its depth in the bit volume. A typical signature will be made of 10 control words for localization, 2 for the grey level and 2 for spatial extension.

3.3. Desiderata of the control function
The choice of the \( f \) function is subject to several requirements:

- \( P1 \): It should be computationally easy to extract \( H \) (or \( h_i \)) from any \( M \) (or \( m_i \));
- \( P2 \): For any signature \( H \), it should be extremely difficult to find \( M \), which implies that a pirate should not be able to reverse the control function.
Figure 4. Segmentation alternatives of the ROI bit volume. Notice that localization can be refined by overlaying two or more segment maps.

- **P3**: For any $M$, it is unlikely to find another message $M'$ such as $f(M') = f(M) = H$ (low collision probability);
- **P4**: For any couple $(H, M)$ it is not possible to calculate $(H', M + \delta M)$, in other words it is not possible to predict the impact of a $M$ modification on $H' = f(M + \delta M)$;
- **P5**: For any $M \simeq M'$ (i.e. two messages with few differences), their signatures $H$ and $H'$ should be very different.

In Section 4 we consider three alternative control functions.

### 4. CONTROL FUNCTION ALTERNATIVES

We consider cryptographic hash functions, parity checking and linear error detecting codes as the three ways of signature extraction. In all cases it is understood that the message is the ROI bits while the control sequence, $H$, which can consist of the message digest or the parity control bits, is to be inserted in the RONI zone.

#### 4.1. Cryptographic hash functions

Cryptographic hash functions\(^{13}\) are commonly used for digital signatures as they extract a resume or digest from the message data. Between the two function classes, the first one, called Message Code Authentication (MCA), uses a secret key and permits signature identification. The second one, known as Manipulation Detection Code (MDC), is calculated without a secret key. Since MCA function usually makes use of a MDC function concatenated with a secret key based encryption algorithm, interest here is given to MDC hash function. These functions are said one way hash functions (i.e. non reversible), and from a message of arbitrary length they provide a fixed length digest or resume.

The best known methods are the SHA (Secure Hash Algorithm) and the MD-5 (Message Digest-5) that yield, respectively, signatures of 160 and 128 bits. They both satisfy properties P1 to P5. In fact their collision probabilities are upper bounded by $1/2^{128} = 2,93 \times 10^{-39}$ and $1/2^{160} = 6,84 \times 10^{-69}$, respectively. They also have good dispersion property in that a slight difference in a message will lead to a very different signature.

Their only shortcoming is that they provide limited spatial resolution, in other words limited ability to localize the alteration(s) in an image. As they output a fixed length signature, the achievable spatial resolution depends on the total insertion capacity, that is the number of 128 or 160-bit control words that can be separately inserted in the peripheral zone. Since the insertion capacity is typically limited to a few hundred bits, this also limits the number of data segments from which separate control words will be extracted to one or two.
4.2. Parity control

The single parity control is the simplest solution\textsuperscript{14} to detect errors. For a word \( m_i \), the corresponding control word \( h_i \) has length \( \eta = 1 \) and corresponds to its parity. Based on exclusive-"or" between the whole segment bits, the parity control is computationally very easy (P1), it is a non-reversible function (P2) and it provides the highest spatial resolution since the parity control can be exercised over \(|H|\) segments of the image data, each of them yielding one control bit. For example the number of segments can be taken as 160 to be commensurate with the hash function. On the other hand the simple parity control does not satisfy any of the properties P3 to P5. Obviously for any \( \mu \)-bit segment there are \( 2^\mu - 1 \) other bit combinations that yield the same parity. The pirate would just toggle an even number bits of \( m_i \) to escape detection or conversely allowing him to predict the change in the signature as a consequence of his manipulations. Such easy manipulations can be foiled, however, by interleaving the message bits in composing the segments. In other words the footprint of the segments on the ROI image plane can be made secret, e.g., using a random permutation as discussed in Section 3.

Notice that if the ROI corresponds to a message size of \(|M|\) bits then each parity bit will be controlling some \(|M|/|H|\) bits. For example if the ROI corresponds to 80 percent of a 512x512 image (let's say \(|M| = 1,600,000\) bits), each parity bit becomes associated with an ensemble of 10,000 bits for \(|H| = 160\). Although the probability of non-detection is practically 50% in any one segment, it is shown in Section 5 that over the ensemble of all segments the simple parity control still becomes a viable solution.

4.3. Linear block codes

Linear block codes are divided in two categories, the group codes defined in a vector space, and the cyclic codes based on polynomial arithmetic.\textsuperscript{15} Here we explore the use of cyclic codes because of their low implementation complexity (P1), and burst error detection.

Linear systematic block codes yield a message \( m \) of \( \mu \) bits concatenated with the control word of size \( \eta \) bits, in total \( n = \mu + \eta \) bits. There are \( 2^\mu \) unique message words in a space of \( 2^n \) code words (P2). The linearity property ensures that the sum of two message words of a code is also a message word. This implies that of the \( 2^{\mu + \eta} \) combinations the \( 2^\mu \) combinations are non-detectable error patterns,\textsuperscript{15,16} while all other error patterns will be detected. For additional secrecy interleaving can be applied to the whole ROI message \( M \) in forming the segments or alternatively on the segments \( m_i \) themselves (see figure 5). Notice that different from the case of single parity check, the bit order has a bearing on the control word. Block codes also do not guarantee the satisfaction of the P3 to P5 constraints. A cyclic code is a linear code block for which any circularly permutation of a code word is also a code word. These codes are specialized for burst errors detection, and constitute the basis of CRC (Cyclic Redundancy Check). The error detecting capability of these codes, such as capturing all bursts of length less than \( \eta \) are well documented.\textsuperscript{16}

The cyclic codes studied here are the Hamming Cyclic Code. For all \( \eta > 3 \), there exists a Hamming code that is able to protect \( \mu = 2^n - \eta - 1 \) information bits, and for which the non-detection probability is upper bounded by \( 1/2^n \) on a BSC (Binary Symmetric Channel) model (ex.: for \( \eta = 18 \), \( 1/2^{18} = 3.8 \times 10^{-6} \)). Therefore the number of

\[ \text{Figure 5. Interleaving of the segments } m_i \text{ to enhance secrecy.} \]
segments, $N$, must, on the one hand, be equal or larger than the number of code words of size $\mu = 2^n - \eta - 1$ to cover the message, and on the other hand, smaller than $N \cdot \eta$, that is the total volume of parity check bits allowed to be embedded:

$$\frac{|M|}{2^n - \eta - 1} \leq N \leq \frac{|H|}{\eta}.$$  

For the given bit volume resulting from the 200,000 pixels of the ROI and for an insertion capacity comparable to that encountered in the message hashes, e.g., 160 bits, it turns out that the message must be partitioned in a few large segments. For example $\eta = 18$ yielding 8 segments seems to be a good compromise. Any surplus capacity (for example, $160 \cdot 144 = 16$ in this particular case) due to the above inequality can be used to improve detection, that is by redistributing them on the control words, by introducing a new segment to be protected with '16' bits, etc.

To improve more this scheme, overlapped segments (i.e. some segments contain common ROI bit) can be exploited such as illustrated in 4.

5. TAMPER DETECTION ANALYSIS

In this section the three control functions are analyzed as for their tamper detection capability. For a concrete example we assume a ROI made up of 75% of image $512 \times 512$ with 8 bit/pixel while signature length is $|H| = 160$ bits.

For the following analysis we need a model for the bit alterations due to any tampering. To this effect a Binary Symmetric Channel (BSC) with crossover probability $p$ is assumed to model for attacks. A BSC model with average probability of crossover is valid because even though a malicious tampering will in any probability be local, it will be dispersed throughout the message sequence due to permutation or interleaving.

1. BSC model for signal processing operations

The bits in the pixel representation are affected in a graduated way depending on the depth of the bit plane and on the nature of the filtering operation. The less significant bits (LSB) are affected more often than the more significant (MSB) ones. In this particular experiment, after a low-pass filtering with a window size $3 \times 3$, it has been observed that almost 50% of the bits in the less significant bit planes do not any more match those in the original image while this BER drops to 0,2 in the most significant bit plane. Similar statistics are also given table 1 for an alternate filtering, that is, sharpening operation. In each case the error probability $\hat{p}$ is estimated as the fraction of the affected bits to the total bit volume $|M|$.

| Table 1. Error probability in a BSC model due to $3 \times 3$ low-pass and sharpening filters. |
|-------------------------------------------|---|---|---|---|---|---|---|---|
| bit planes LSB to MSB                      | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | $\hat{p}_{mean}$ |
| Averaging filter $\hat{p}_{ave-plane}$     | 0.48| 0.35| 0.22| 0.12| 0.07| 0.03| 0.01| 0.002| 0.16 |
| Sharpening filter $\hat{p}_{sh-plane}$     | 0.5 | 0.5 | 0.49| 0.41| 0.27| 0.13| 0.07| 0.01 | 0.29 |

2. BSC model for tampering consisting of replacement of a local region

In this case the attack consists of a malevolent alteration, such as adding or removing a lesion artifact locally in the ROI. If the pirate tampers with the image by replacing a zone, the BSC error probability $\hat{p}$, is simply proportional to the area of the manipulated zone. Although the altered pixels are in general correlated, the fact of scanning the bits across the planes and the segmenting of the image (the construction of the message as in Figs. 3 and 4) result in an uncorrelated sequence and the BSC model is conjectured to be applicable. Table 2 gives some estimation of $p$ for all the ROI.

5.1. Tamper detection with hash functions

The SHA function protects all the ROI as one entire segment. The probability of non-detection, that is of another image passing for a valid one, is less than $1/2^{100} = 6,84.10^{-49}$. However no localization of the tamper is possible.
Table 2. BSC p error probability due to the quantity of tampered pixels in a 140000-pixel ROI.

<table>
<thead>
<tr>
<th>Quantity of altered pixel</th>
<th>1</th>
<th>11</th>
<th>1400</th>
<th>22400</th>
<th>42000</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{p} )</td>
<td>0.000005</td>
<td>0.00008</td>
<td>0.01</td>
<td>0.16</td>
<td>0.3</td>
</tr>
</tbody>
</table>

5.2. Tamper detection with single parity bit

Any manipulation that alters an even number of bits will go undetected. The non-detection probability \( P_{nd} \) on an image segment represented by a word \( m \) of length \( \mu \) bits, for a BSC model with parameter \( p \), can be written as:

\[
P_{nd}(\text{segment}) = \sum_{k=1}^{\mu/2} C_\mu^{2k} p^{2k} (1 - p)^{\mu - 2k},
\]

and as in our concern \( \mu \) is rather large, using the Poisson Distribution, approximated by:

\[
P_{nd}(\text{segment}) \approx \frac{1}{2} (1 - e^{-\mu p})^2.
\]

Obviously this probability approaches 0.5 for non-negligible values of \( p \). As shown in Fig. 6, the non-detection probability remains low only for \( p < 10^{-4} \), which for a segment size of \( \mu_i = 10,000 \), corresponds to only one or two pixel alterations. Though single parity checking provides insignificant protection to any one segment, the combinatorial probability of non-detection over all the segments is much lower and is calculated by considering the probability that a segment is altered while not being detected:

\[
P_{nd}(\text{ROI}) \approx \sum_{b=1}^{N} C_N^{\mu} [1 - (1 - p)^b] (1 - p)^{\mu (N - b)} \left( \frac{(1 - e^{-\mu p})^2}{2} \right)^b.
\]

Again the underlying assumption is that the total bit volume is split into 160 segments, each protected by a single bit, but that any tampering effect is assumed to 'spread' to all the segments of the image via interleaving or permutation of the bits. For \( p > 0.0001 \) (160 tampered bits on a 1,600,000-bit ROI), all segments are affected and leads to a performance equivalent to that of the SHA function. It can be observed that for sufficient \( p > 0.0001 \) (and \( \mu = 10,000 \)) their performances are similar. On the other hand for \( p \) less than this value, the non-detection probability of the single parity check scheme becomes worse \((P_{nd}(\text{ROI}) \approx 2.6 \times 10^{-5})\) as it loses the combinatorial advantage since there will be an increasing number of non-affected segments for vanishing \( p \).

**Figure 6.** Non-detection error probability over a segment with the single parity check technique \((\eta = 1, \mu = 10,000)\) versus the BSC error rate \( p \).
5.3. Tamper detection with Hamming codes

The non-detection (miss) probability of a Hamming code\textsuperscript{17} when $\eta$ parity bits are used to protect a segment is given as a function of BSC probability $p$ by:

$$P_{nd}(\text{segment}) = 2^{-\eta}\left\{1 + (2^\eta - 1)(1 - 2p)^{2^{\eta - 1}}\right\} - (1 - p)^{2^{\eta - 1}}.$$

If we again consider numerical example, within the signature length of 160 bits, using the inequality proposed section 4.3, one can accommodate 8 18-bit long control words, each for the control of one eighth of the message $M$. Alternatively one can use 8 20-bit Hamming code to use all of the 160 bits.

Therefore the image bit volume is organized as 8 segments $\mu = 196,628$ bits long. Notice that for a $(n, \mu, \eta)$ Hamming code words, with $\eta = 18$ the source word (in our concern the segment), can be as long as 262,125 bits and as 1,048,555 bits if $\eta = 20$. This free space can be used to improve detection performance (repetition of some bits), ignored or used to protect another information.

It is interesting to compare the performance of the Hamming and single parity schemes. Figure 7 shows the performance over a 262,143 bit segment protected by a $(262143, 262125, 18)$ Hamming code against that of the single parity scheme organized as 18 subsegments each one containing one 18th of the bits. Notice again that for non-negligible probability of attack their performance is comparable, while the Hamming scheme can detect tampering at arbitrarily low $p$ probability. Note that at higher $p$, when there are more than 18 errors, Hamming will be failing while single parity can still function due to its combinatorial strength.

The probability of non-detection with the Hamming code over all the $N$ segments of the image, again under perfect segment interleaving assumption, becomes:

$$P_{nd}(\text{ROI}) = \sum_{l=1}^{N} C_N^l \left[1 - (1 - p)^\mu\right]^\phi (1 - p)^{\mu(n-l)}P_{nd}(\text{segment})^l.$$

This performance is plotted in Fig. 8 (8 segments each with 18 parity bits) where one can observe that the tamper detection performance is at a comparable level with that of the SHA for $p > 5.10^{-5}$. However for lower $p$ values this performance drops down to $0.5 \cdot 10^{-8}$ due to the fact lesser and lesser number of segments become affected. To give an idea of the extent of tampering, a $p$ value of $10^{-5}$ corresponds to about 4 altered pixels, if one assumes that when a pixel is modified, half of its bits will differ. Such a tampering is of course semantically not very relevant.

Figure 9 gives the comparison between the SHA non-detection probability upper bound and a system based on 8 segments each protected with a 20-bit Hamming code. For high $p$, the SHA performance are reached. Using the overlapped segments as illustrated on figure 4 could improve these results as for low and high $p$ values.

**Figure 7.** The $(1031, 1013, 18)$ Hamming code (point) performance versus the single parity bit scheme (solid) consisting of 18 segments each containing 102 bits.

Table 3 summarizes on a comparative basis the characteristics of the three methods considered, namely, the SHA, the single parity check and the Hamming code.
Figure 8. Image tampering non-detection probability using 8 Hamming code words (262243, 262125, 18).

Figure 9. Image tampering non-detection probability using (1048575,1048555, 20) Hamming codes on 8 segments (solid) versus the SHA upper bound (point) for a final signature of 160 bit long.

6. CONCLUSION

The role of watermarking in the context of tamper protection of medical images has been considered. The solution offered is specific to medical images in that they can be segmented into two zones, the Region of Interest, whose integrity is to be protected, and a Region of Non-Interest, serving for watermark insertion. The signature extracted from ROI is to be inserted in the RONI, where the imperceptibility constraint is less demanding in providing desired capacity and robustness.

Three feature extraction schemes from the ROI to constitute the segment content have been proposed. Depending on their use, i.e. independently or together, the nature of the ROI alteration can be surrounded. A typical example for 14 control words (see section 3.2) of 18 bits each leads to a final signature of 252 bit long.

We have comparatively analyzed three signature calculation techniques from the ROI bit field, namely, the hashing functions, the single parity control and the error control codes. It has been shown that hashing functions satisfy not only the requirements cited in section 3.2, but also provide the highest protection, that is the highest probability of detecting any alteration, but they lack spatial localization because of their irreducible digest or signature length as hashing functions process the whole image bit volume in 'one go'. Signatures obtained from error detection codes allow processing of the image in several blocks, hence provide some localization ability. As these codes are well known, secrecy is obtained through the use of permutation of the bit positions on all the ROI or on the segments separately. For single parity control to be effective the permutation should be used on the whole ROI.

Under bit permutation assumption, any alteration could be modeled as a BSC with an error probability p. Under this model, in the case of a final signature $H$ of $|H| = 160$ bit long, SHA, Hamming codes and single parity control have commensurate performance when the ROI alteration probability is above a minimum threshold of $5 \times 10^{-5}$. Below
Table 3. A comparison of the three control functions

<table>
<thead>
<tr>
<th>Feature</th>
<th>SHA</th>
<th>Single parity control</th>
<th>Hamming code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity (P1)</td>
<td>high</td>
<td>low</td>
<td>medium</td>
</tr>
<tr>
<td>Non-reversibility (P2)</td>
<td>high</td>
<td>high</td>
<td>high</td>
</tr>
<tr>
<td>Collision (P3)</td>
<td>2η</td>
<td>2η-1</td>
<td>in average 2η-1</td>
</tr>
<tr>
<td>Prediction (P4)</td>
<td>high</td>
<td>ROI interleaving needed</td>
<td>segment interleaving needed</td>
</tr>
<tr>
<td>Dispersion</td>
<td>high</td>
<td>low</td>
<td>low</td>
</tr>
<tr>
<td>Control word length (η)</td>
<td>160</td>
<td>1</td>
<td>η &gt; 3</td>
</tr>
<tr>
<td>Maximum segment size (μ)</td>
<td>128 Gigabits</td>
<td>unlimited</td>
<td>2η - η - 1</td>
</tr>
<tr>
<td>Segment non detection probability</td>
<td>&lt; 0.5^100</td>
<td>&lt; 0.5</td>
<td>&lt; 0.5^100</td>
</tr>
<tr>
<td>ROI lowest/highest non detection probability (</td>
<td>0.5^100/0.5^100</td>
<td>2.5.10^-6/0.5^100</td>
<td>1.10^-10/0.5^100</td>
</tr>
</tbody>
</table>

this threshold parity checking and/or error detecting codes may fail to detect tampering. However such low BSC error rates also correspond to very few pixel alterations, less than 10, which can hardly correspond to a piracy.

In this respect block codes seem to be the best compromise especially if the capacity is limited. If coupled with interleaving of bits and overlapping of segments (partitions of ROI) they also provide adequate protection against collusion attack and a modest localization of tampering.

REFERENCES