Chapter 6

Multilevel Access to Precision-Critical Data

When protecting data in the sense addressed in this thesis (transparent protection), the protection method introduces some amount of noise into the data. For statistical microdata, this noise corresponds to the distortion caused by a statistical disclosure control (SDC) method; for multimedia content, noise is caused by the modifications performed on the multimedia content in order to embed a watermark (or fingerprint).

Generally, the noise injected by transparent protection is quite tolerable. A masking method applied over a microdata file must not substantially alter the statistical properties of the file. In the same sense, if the mark embedding algorithm used for watermarking satisfies the property of imperceptibility, the marked multimedia content has the same commercial value as the original one. However, when dealing with precision-critical data, any small amount noise may render the data useless for some applications.

In this chapter we address this problem and propose a solution for *multilevel access to precision-critical data.* The goal of our solution is to provide multilevel access to precision-critical data, in such a way that:

- Non-privileged users just see the protected data
- The higher the clearance of the user, the more protection she can remove:
  - In SDC protected microdata, such protection removal translates to partial removal of the distortion, and thus to data more similar to the original.
  - In watermarked multimedia contents, such protection removal allows the unwatermarked version of some parts of the content to be recovered.

6.1 Watermarking for multilevel access to statistical databases

A novel application of watermarking is presented in this section which allows multilevel access to numerical microdata: depending on her clearance, the data user can remove more or less of the masking. Non-privileged users just see the published data, but, as the clearance of a user increases, she can get a data set which is closer and closer to the original one. Results presented in this section have been published in [DMS01].

6.1.1 Partially removable masking

We assume that the information of a microdata file is represented as a two-dimensional table where one dimension corresponds to the set of objects (*i.e.* elements, individuals, persons) and the other is the set of attributes (*i.e.*

variables). The microdata file contains a value for each object-attribute pair, so that it can be modelled as a function

$$X : O \to D(X_1) \times D(X_2) \times \cdots \times D(X_d)$$

where O denotes the set of objects, $X_1, X_2, \cdots X_d$ denote the attributes and $D(X_i)$ refers to the domain of attribute X_i. Without loss of generality, the d-dimensional function X can be assumed to be of the form:

$$X(\cdot) = (X_1(\cdot), X_2(\cdot), \cdots, X_d(\cdot))$$

where $X_i(\cdot) : O \to D(X_i)$ is a one-dimensional function assigning a value for attribute X_i to a given object.

Assumption 1 We assume that a perturbative masking method can be expressed as a masking algorithm F which takes as inputs the original microdata file X and the outputs of r pseudorandom number generators $PRNG_i$ seeded by s_i, for $i = 1, \cdots, r$. The output of F is the masked microdata file X'. Formally speaking,

$$X' = F(X, \{s_1, \cdots, s_r\})$$

F and $PRNG_i$, for $i = 1, \cdots, r$ are assumed to be public, so the only secret parameters of masking are s_i, for $i = 1, \cdots, r$.

Assumption 2 We assume that each $PRNG_i$ is used to independently mask a part of the microdata file, so that *knowledge of the random numbers generated by $PRNG_i$ should allow to retrieve the original values from the corresponding masked values in that part of the microdata file.* Formally speaking, given a subset $S \subset \{s_1, \cdots, s_r\}$, we can compute

$$X'(S) = F^{-1}(X', S)$$

where $X'(S)$ is a microdata file resulting from removing the masking of X' that was produced using generators seeded by elements in S. In particular $X'(\{s_1, \cdots, s_r\}) = X$.

For some masking methods, it is easy to meet Assumptions 1 and 2, because they make explicit use of random number generation and knowledge of the generated random numbers suffices to undo the masking. Such is the case for additive noise.

For methods which do not directly meet both assumptions above, consider the sequence of differences between the masked and the original data

$$X_i'(o_j) - X_i(o_j) \text{ for } i = 1, \cdots, d \text{ and } j = 1, \cdots, n \tag{6.1}$$

where d is the number of attributes and n is the number of objects.

Now, the Berlekamp-Massey algorithm [Mas69] can be used to synthesize a Linear Feedback Shift Register (LFSR) generating the sequence (6.1). More generally, r LFSRs can be synthesized such that their interleaved outputs yield the sequence (6.1) (the i-th LFSR generates integers in positions j of the sequence such that $j \bmod r = i$). This construction reduces any perturbative method to a variant of additive noise (X' can be computed by adding the Sequence (6.1) to X), which thus meets Assumptions 1 and 2.

Now, if a user is revealed a subset S of the seeds, by Assumption 2 she can remove the masking in those parts of X' masked using generators seeded by values in S, so that the user obtains a partially unmasked file $X'(S)$. In particular, if the user is revealed all seeds, she can retrieve the original file

$X'(\{s_1, \cdots, s_r\}) = X$.

6.1.2 Watermarking solutions for multilevel access

From the previous section, we can see that, the larger the subset S of seeds known by a user, the more masking the user can remove, *i.e.* the closer is the unmasked file $X'(S)$ to the original X. This suggests the following algorithm to implement multilevel access to the masked file X':

Algorithm 14 *1. Let H be a clearance hierarchy comprising u user categories (for example, "statistician", "researcher", "civil servant", "other users"). For each category j, let k_j be a secret key known only to users in that category (the user does not actually need to know k_j, which can reside in her smart card).*

2. For $i = 1, \cdots, r$ and $j = 1, \cdots, u$, encrypt s_i with some redundancy R_i under k_j to get $E_{k_j}(s_i||R_i)$ if s_i should be revealed to user category j.

3. Use a watermarking algorithm to embed $E_{k_j}(s_i||R_i)$, for $i = 1, \cdots, r$ and $j = 1, \cdots, u$, into the masked file X' to get a watermarked file X''.

From X'', a user can retrieve the subset S of seeds her category is entitled to know, and thus retrieve $X'(S)$. Redundancy R_i encrypted with s_i allows the user to check that s_i was correctly decrypted. We next discuss which features the watermarking algorithm should offer.

6.1.3 Watermarking requirements

Unlike in most usual watermarking applications (see [CMB00]), watermarking in the application described here is positive for the user. In the worst case, the user with no clearance just gets X'', but the user with some clearance gets a better file. Therefore, there is no reason to expect a malicious behaviour by the user to destroy the watermark. Robustness should provide for those normal accidental alterations that may occur during the life cycle of a numerical file. These are basically rounding errors, mostly due to the software used to manipulate the data (*e.g.* when importing an ASCII version of X'' into a spreadsheet which rounds to two decimal positions). The rest of processing manipulations an image watermark should resist (see [PAK98]) do not make much sense on a microdata file, because nobody is really interested in a cropped, scaled or compressed version of X''.

The capacity of the watermarking scheme should be sufficient to allow embedding of $E_{k_j}(s_i||R_i)$, for $i = 1, \cdots, r$ and $j = 1, \cdots, u$. It must be noticed that a numerical microdata file is usually smaller than multimedia files: just in one color 512×512 RGB image, we have 3×2^{18} pixel values, which is more than the number of values in a typical microdata file. Thus capacity should be medium to high in comparison to standard multimedia watermarking schemes.

Regarding obliviousness and imperceptibility, there is an interesting tradeoff in this multilevel access application:

- An oblivious watermarking scheme does not require X' to recover the watermark from X''. In principle, distribution of X' is thus unnecessary, which saves storage and communication. However, this means that the user will remove masking from X'' rather than from X'; so unless both files are very similar (*i.e.* the watermark is very imperceptible), unmasking X'' will not yield analytically valid results, because masking was performed on X'.
- A non-oblivious watermarking scheme assumes that X' is available when recovering the mark from X''. So there is no problem if X''

differs significantly from X', because the user will be able to perform the unmasking on X'. Thus, for a non-oblivious watermarking scheme, imperceptibility is not a requirement.

6.1.4 Choice of a watermarking algorithm

As noted in Section 2.3.1 when discussing SDC methods based on lossy compression, a numerical microdata file can be regarded as an image. Therefore, all image watermarking algorithms are potentially usable. However, from the requirements analysis of Section 6.1.3, we can conclude the following:

- Robust oblivious schemes like [HG96] cannot yield enough capacity while preserving a good level of imperceptibility. For example, for a microdata file with 1080 records and 13 variables, 10% distortions are needed to embed a 60-bit mark using [HG96]; it can be empirically seen that the amount of bits that can be embedded grows linearly with the percent distortion being used. In spite of 10% being already a distinctly perceptible distortion, a 60-bit mark can hardly accomodate a single encrypted seed (whereas embedding several encrypted seeds would be desirable).

- Robust non-oblivious schemes like [Her00, SDH00] offer a good level imperceptibility and good capacity, but require distributing X' along with X''.

- Oblivious methods, like Least Significant Bit (LSB) embedding, which are not robust enough for image watermarking, may be successfully adapted for the purposes of the application discussed here. Basically, the only manipulation LSB methods should survive in our case is quantization (due to rounding errors): this can be achieved by embedding one bit in a *group* of least significant bits rather than in the least significant bit. LSB methods offer high imperceptibility and allow embedding one bit in each numerical value of the microdata file, so they offer high capacity as well.

6.2 Multilevel access to precision-critical images

A novel application of invertible watermarking is presented in this section which allows multilevel access to precision-critical images. Our goal is to devise a mechanism for user access to precision-critical images whereby the user can invert the embedded watermarks (and thus the distortion they cause) to an extent proportional to her clearance. Users with no clearance at all only see the watermarked image, which is copyright-protected and perceptually good but not suitable for high-precision processing. At the other end, users with full clearance can completely invert the watermarking process so as to obtain the original precision-critical image from the watermarked one. Between both extreme user types, users with intermediate clearances can invert the watermarking for some parts of the image. Results presented in this section have been published in [DS02b].

6.2.1 Mark embedding for multilevel access

Next we present an algorithm to implement multilevel access to precision-critical images.

Assumption 3 *We assume that the image to be protected can be divided* *into r semantically significant disjoint subimages. The i-th subimage is watermarked using an invertible method keyed with a different seed s_i. Formally speaking, if we denote the original image by X, the copyright sequence by M, the watermarked image by X' and the watermarking transformation by F, we have*

$$X' = F(X, M, \{s_1, \cdots, s_r\})$$

Under the above assumption, knowledge of s_i allows the original i-th subimage to be retrieved from its watermarked version. More formally, given a subset $S \subset \{s_1, \cdots, s_r\}$, we can compute $X'(S) = F^{-1}(X', S)$, where $X'(S)$ is a partially unwatermarked image resulting from inverting the watermarks in the subimages of X' that were watermarked with seeds in S.

From the previous discussion, we can see that, the larger the subset S of seeds known by a user, the more watermarks the user can invert, *i.e.* the closer is the unwatermarked image $X'(S)$ to the original X. This suggests the following algorithm to implement multilevel access to the watermarked file X':

Algorithm 15

1. *Let CH be a clearance hierarchy comprising u user categories (for example, for medical images, we could think of "doctor", "nurse", "other users"). For each category j, let k_j be a secret key known only to users in that category (the user does not actually need to know k_j, which can reside in her smart card).*

2. *For $i = 1, \cdots, r$ and $j = 1, \cdots, u$, encrypt s_i with some redundancy R_i under k_j to get $E_{k_j}(s_i||R_i)$ if s_i should be revealed to user category j.*

Note that different seeds may be used for each image, whereas the key k_j corresponding to category j is assumed to stay stable.

3. *Assuming that the invertible watermarking algorithm used allows multiple marking without significant increase of the non-invertible distortion, embed $E_{k_j}(s_i||R_i)$, for $i = 1, \cdots, r$ and $j = 1, \cdots, u$, into the watermarked file X' to get a rewatermarked file X''. A public seed is used for this second watermarking round.*

6.2.2 Partial mark removal

From X'', a user can recover and decrypt the subset S of seeds her category is entitled to know together with X', and thus retrieve $X'(S)$. Redundancy R_i encrypted with s_i allows the user to check that s_i was correctly recovered and decrypted.

As pointed out in Section 3.5, the Hartung-Girod method is an example of invertible watermarking which supports multiple marking. The only shortcoming of using n marking rounds (as required by Step 3 of Algorithm 15) is that α in the pre-processing algorithm (Algorithm 10) must be replaced with $n\alpha$. Thus, two marking rounds result in an increase of the non-invertible distortion introduced at the pre-processing stage. An alternative to avoid embedding $E_{k_j}(s_i||R_i)$ in the image is to keep those encrypted values in a freely accessible public repository.

Example: Figure 6.1 shows the original image "Chips" [1] and its division into 12 subimages. The Hartung-Girod method has been used to embed the same watermark in each subimage.

[1] http://www.microscopy.fsu.edu/micro/gallery/chips/chipshots.html

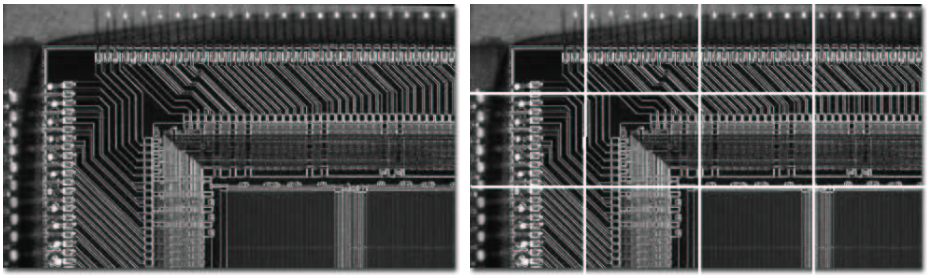

Figure 6.1: Left, original Chips image. Right, subimage division of Chips (12 tiles).

A partially unwatermarked version offers maximum precision and integrity verification to a user wishing to inspect the chip contact area depicted in the unwatermarked subimages; the remaining subimages showing the rest of the chip still carry watermarks whose copyright messages can be used to prove ownership in case of unlawful redistribution.