2
Preliminaries

2.1 Data Hiding Framework

A typical data hiding framework is illustrated in Fig. 2.1. Starting with an original digital media (I_0), which is also commonly referred as the *host media* or *cover media*, the embedding module inserts in it a set of secondary data ($\underline{b}$), which is referred as *embedded data* or *watermark*, to obtain the *marked media* (I_1). The insertion or embedding is done such that I_1 is perceptually identical to the original I_0. The difference between I_1 and I_0 is the distortion introduced by the embedding process and is referred to as *embedding distortion*.

In most cases, the embedded data is a collection of bits, which, depending on the application, may come from an encoded character string, from a pattern, from some executable agents, or other sources. For such generic hidden data, we concern the bit-by-bit accuracy in extracting them from the marked media. The embedded data may also form a perceptual source, such as the application of "image in image" and "video in video" [104, 103]. Some moderate decay of the hidden data is tolerable in this case.

The embedded data b is to be extracted from the marked media I_1 by a detector, often after various processing and attacks. The input media to the detector is referred to as *test media* (I_2), and the difference between I_2 and I_1 is called *noise*. The *extracted data* from I_2 is denoted by $\hat{\underline{b}}$. In such applications as ownership protection, fingerprinting / recipient tracing, and access control, accurate decoding of hidden data from distorted test media is preferred. They are commonly referred as *robust data hiding /*

watermarking. In other applications such as authentication and annotation, robustness against processing and attacks are not a principal requirement in general. We will discuss the specific design requirement in the later chapters.

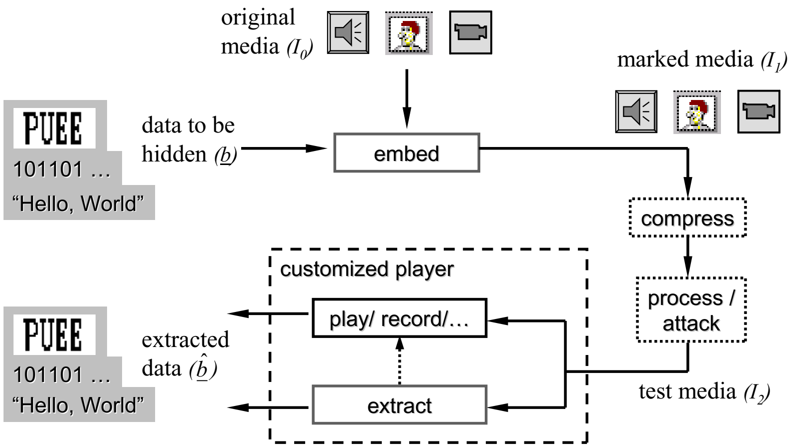
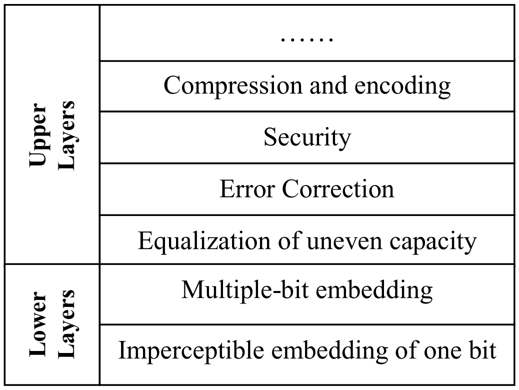


FIGURE 2.1. General framework of data hiding systems



Upper Layers	……
	Compression and encoding
	Security
	Error Correction
	Equalization of uneven capacity
Lower Layers	Multiple-bit embedding
	Imperceptible embedding of one bit

FIGURE 2.2. Layered structure of a data hiding system.

2.2 Key Elements and A Layered View

The key elements in many data hiding systems include [82]:

1. A perceptual model that ensures imperceptibility;
2. How to embed one bit;
3. How to embed multiple bits via modulation/multiplexing techniques;
4. How to handle the parts of host media in which it is difficult to embed data, or more generally, how to handle uneven embedding capacity;
5. How to enhance robustness and security;
6. What data to embed.

These elements can be viewed in layers (Fig. 2.2), analogous to the layered structure in network communication [9]. The lower layers deal with how one or multiple bits are embedded imperceptibly in the original media. The three related key elements are: (1) the mechanism for embedding one bit, (2) the perceptual model to ensure imperceptibility, and (3) the modulation/multiplexing techniques for hiding multiples bits. Upper layers for achieving additional functionalities can be built on top of these lower layers, for example, to handle uneven embedding capacity, to enhance robustness and approach capacity via error correction coding, and to incorporate additional security measures. In the remaining chapters of Part I, we shall use data hiding in images as an example to discuss several elements in detail.