

Attacks on Unknown Data Hiding Algorithms

In last section, we discussed watermark attacks with the embedding and detection algorithms known to analysts, which is the case for most attacks studied in literature. The recent public challenge organized by the Secure Digital Music Initiative (SDMI) provided a research opportunity to study attacks under an emulated rivalry environment. We have proposed successful attacks on all four watermarking schemes currently under SDMI's consideration, pointing out the weaknesses and proposing some directions of improvements. We have also found a few general approaches that would be used by an attacker in a real rivalry environment and demonstrated a framework for studying the robustness and security of data hiding systems.

10.1 Introduction

Secure Digital Music Initiative (SDMI) is an international consortium that is developing open technology specifications aiming at protecting the playing, storing, and distributing of digital music [140]. Imperceptible digital watermarking has been proposed to be key elements in the SDMI systems. Upon detection, the watermarks

may direct certain actions to be taken, for example, to permit or to deny recording. An SDMI system may incorporate a combination of robust and fragile watermarks. Robust watermarks can survive common signal processing and attacks and are crucial for ensuring the proper functioning of the entire system. The fragile watermarks may be used to indicate whether the audio has experienced certain processing such as lossy compression [141]. The SDMI watermarks are considered as *public watermarks*, meaning that (1) the detection does not use the original unwatermarked copy (i.e., blind detection), and (2) a single or a set of secret keys for detecting the watermarks are usually encapsulated in all publicly available detection devices. In early September 2000, SDMI announced a three-week public challenge for its Phase-II screening, inviting the public to evaluate the attack resistance for four watermark technologies (A, B, C, F) and two other technologies (D, E). In the following, we summarize the attacks and analysis on four watermark technologies.

10.1.1 SDMI Attack Setup

In this challenge, the watermark embedding and detection algorithms are not known to the public. Limited information is available only through the oracle submission. After each submission, the detection is performed by the SDMI staff and the result is sent back with a response time of about 4 ~ 12 hours. For each of the four challenges, SDMI provided three audio samples, as illustrated in Fig. 10.1. They are:

- samp1?.wav (original audio with no watermark)
- samp2?.wav (samp1?.wav watermarked by Technology-?)
- samp3?.wav (a different audio watermarked by Technology-?)

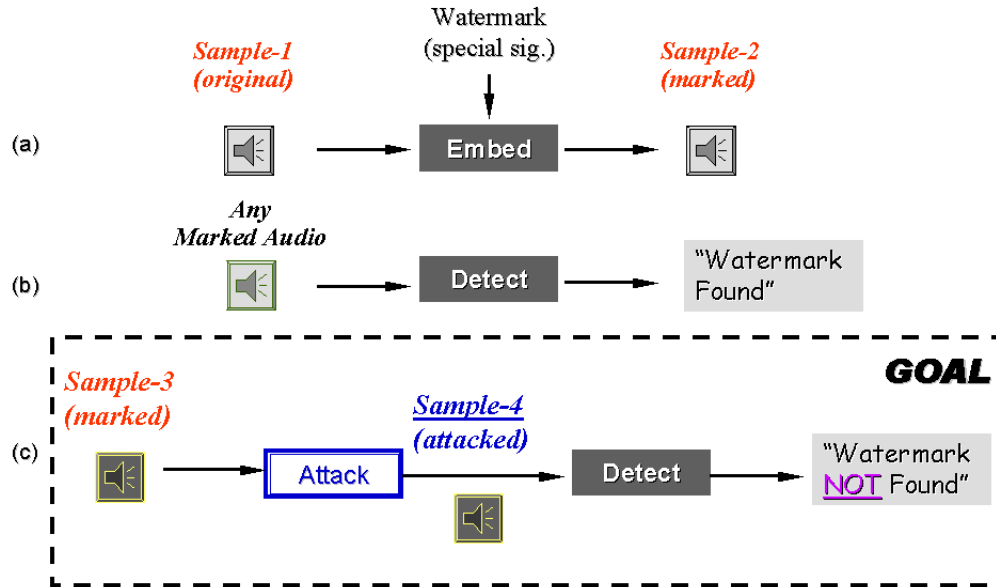


Figure 10.1: Illustration of SDMI attack problem. For each of the four watermark challenges, Samples 1 ~ 3 are provided by SDMI. Sample 4 is generated by participants in the challenge and submitted to SDMI oracle for testing.

where the substitution symbol “?” stands for one of the four challenges: “a”, “b”, “c”, or “f”. All audio samples are 2-minute long, sampled at 44.1 kHz with 16-bit precision. The audio contents are mostly popular music. Sample-1 for all four technologies are identical, while sample-3 are all different.

A participant of this challenge generates an attacked audio file *sample-4* from *sample-3*, then uploads it to SDMI’s oracle for testing. The detection response is binary, i.e., either “possibly successful” or “unsuccessful”. According to SDMI’s emails, a “possibly successful” attack must render the detector unable to find the watermark, while retaining the auditory quality comparable to the original one (*sample-3*). This indicates that a successful attack should sit in the region IV of Fig. 10.2. Interestingly, in the unsuccessful case, there is no indication whether the detector can still

find watermark (region II of Fig. 10.2) or the detector can no longer find watermark but the auditory quality is considered unsatisfactory (region III of Fig. 10.2). For convenience, we shall denote the four pieces of audio as S_1 , S_2 , S_3 , and S_4 .

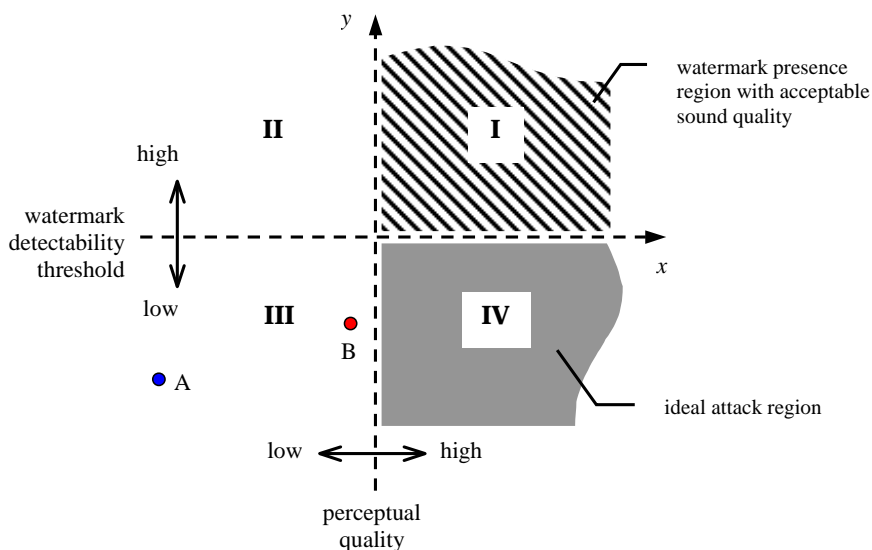


Figure 10.2: Illustration of watermark detectability and perceptual quality .

10.1.2 Comments on Attack Setup

The SDMI public challenge presents an emulated rivalry environment, providing attackers with a limited amount of information and restricted access to watermark detectors in a very short time frame. The task is more difficult than the one in real world. First, in real world, a watermark detector encapsulated in a compliant device will be available to an attacker for unlimited uses, and the detector's response time will be instantaneous rather than hours. Second, a user of the real system will be able to distinguish whether or not a detector is able to find watermarks, regardless

of the audio quality. These two aspects would enable an attacker polling the detector with different input and obtaining the corresponding output, which in turn provides a large amount of useful information for attacks. Furthermore, the SDMI business model allows a user to pass a piece of non-SDMI music that does not have watermark embedded through an SDMI admission device to make it SDMI-compliant thus has watermark embedded in. This implies that a non-trivial number of original-watermarked audio pairs rather than a single pair are likely to be available to an attacker in real world. As can be seen in the next section, these pairs provide valuable information regarding how watermarks are embedded and the information can be exploited in attacks. One should also note that the perceptual quality imposed on embedding and on attacks are different in reality. The quality criterion for embedding is much higher because part of the commercial value of a piece of audio is determined by the sound quality and in many situations it has to meet the most critical demand among a highly diversified audience (from easy listening by the general public to the professional listening by the experts). On the other hand, the sound quality criterion for attacks only need to satisfy a less demanding audience who are willing to tolerate slightly poor quality if for no-fee listening.

The setup also suggests that the SDMI challenge emphasized on evaluating the effectiveness of robust watermark in each technology and did not take much consideration on the fragile watermark. Referring to SDMI's business model, to enforce a copy control policy that allows no MP3 compression on a piece of music prior to the admission to an SDMI compliant device, the robust watermark embedded in the music would convey to the device this policy while the fragile watermark will be used to detect whether the music experiences compression or not. If the bits in the fragile watermark are designed to be a pre-determined secret pattern and are independent

of the host audio, an attacker may obliterate the above policy by restoring a fragile watermark after performing MP3 compression. This attack is likely to introduce less perceptual distortion than removing a robust watermark, therefore, should be given sufficient consideration. The fragile watermarking can be formulated as an authentication problem, for which the attacks and counter-attacks can be studied similar to the material in Chapter 7 and Chapter 9. In the following, we first report our attacks and analysis on the robust watermark in SDMI challenge, then briefly discuss issues related to the fragile watermark.

10.2 Proposed Attacks and Analysis on SDMI Robust Watermarks

In this section, we first explain a general framework for tackling the attack problem. We then take two different successful attacks on Watermark **C** as examples to demonstrate our attack strategies, to describe the specific implementation, and to present analysis in detail. For completeness, the attacks for the other three watermark techniques **A**, **B**, and **F** are also briefly explained.

10.2.1 General Approaches to Attacks

An attacker may take one of three general approaches to tackle the problem: (**Type-1**) exploiting the design weakness via blind attack, (**Type-2**) exploring the embedding mechanism from $\{S_1, S_2\}$, the known original-watermarked pairs, or from the watermarked signal $\{S_3\}$ alone, (**Type-3**) a combination of the two.

Type-1 attacks are said to be *blind* in the sense that they do not rely on any understanding of embedding mechanism or the special properties held by watermarked

signals. This approach includes commonly used robustness tests, such as compression, time-domain jittering, pitch change, resampling at different rate, D/A-A/D conversion, and noise addition [142]. The counter-attack strategy for such blind attacks is to find as many weaknesses as possible and to correct them. A good design, therefore, should at least have covered most of the typical robustness tests and their combinations. One of our attacks for Watermark-C and our attack for Watermark-F are blind attacks.

Type-2 attacks are designed using the knowledge about the embedding mechanism. Such knowledge, even if not available at the start, can be obtained by studying the input-output response of the embedding system. For example, if we find the difference between S_1 and S_2 is a small signal around certain frequency, we may design an attack to distort S_3 over the corresponding frequency range. Quite a few of our attacks belong to this category. This type of attack is analogous to the plaintext attack or ciphertext attack in cryptanalysis ¹ [11]. The differences are: (1) signal processing analysis replaces the cryptanalytic tools in creating watermark attacks, and (2) the goal of watermark attacks is to render detector unable to detect the watermarks, instead of “cracking codes”. The useful signal processing tools include the time-domain and frequency-domain differences, the frequency response, the auto- and cross-correlation, and the cepstrum analysis [12]. We also note that the original and watermarked signals are not easily available simultaneously to the public in some watermarking or data hiding applications, e.g., watermarked-based authentication or DVD video watermarking system. Hence, Type-2 attacks may not be a major concern in those cases. But in SDMI applications where an unwatermarked music

¹*Plaintext attack* refers to deducing the encryption key or decrypting new cipher texts encrypted with the same key, based on the cipher text of several messages and their corresponding plaintext. *Ciphertext attack* only uses the knowledge of the cipher text of several messages.

may be “admitted” into SDMI domain by embedding a watermark, any successful watermarking design has to take Type-2 attacks into consideration. One possible counter-attack strategy is to intentionally wipe off the otherwise distinct “signature” of a particular embedding. Some obscuring processes may reduce the robustness against blind attacks if the obscuring distorts the embedded watermarks, showing a tradeoff among robustness against various attacks.

Because it is not always possible to find clear clues about embedding from a limited number of original-watermarked pairs, especially when the “wipe-off” is applied, attacks can be designed by combining the above two.

10.2.2 Attacks on Watermark-C

We have proposed two different attacks on Watermark-C. *Attack-C1* explores the weakness of Watermark-C under pitch change. *Attack-C2* is based on observing the difference between original and watermarked signal $\{S_1, S_2\}$. Both attacks were confirmed as successful by SDMI oracle.

Observations from Samples of Watermark-C

By taking the difference of `samp1c.wav` and `samp2c.wav`, bursts of narrow band signal are observed, as shown in Fig. 10.3. These bursts appear to be around 1350 Hz.

Attack-C1

Attack-C1 accelerates audio samples by a small amount, which in turn changes the pitch. Blind attacks of 3% pitch increase have been applied to all four watermark proposals, and SDMI detectors indicated that they are effective to Watermark C. The relations between the input and output time index of this speed-up is illustrated

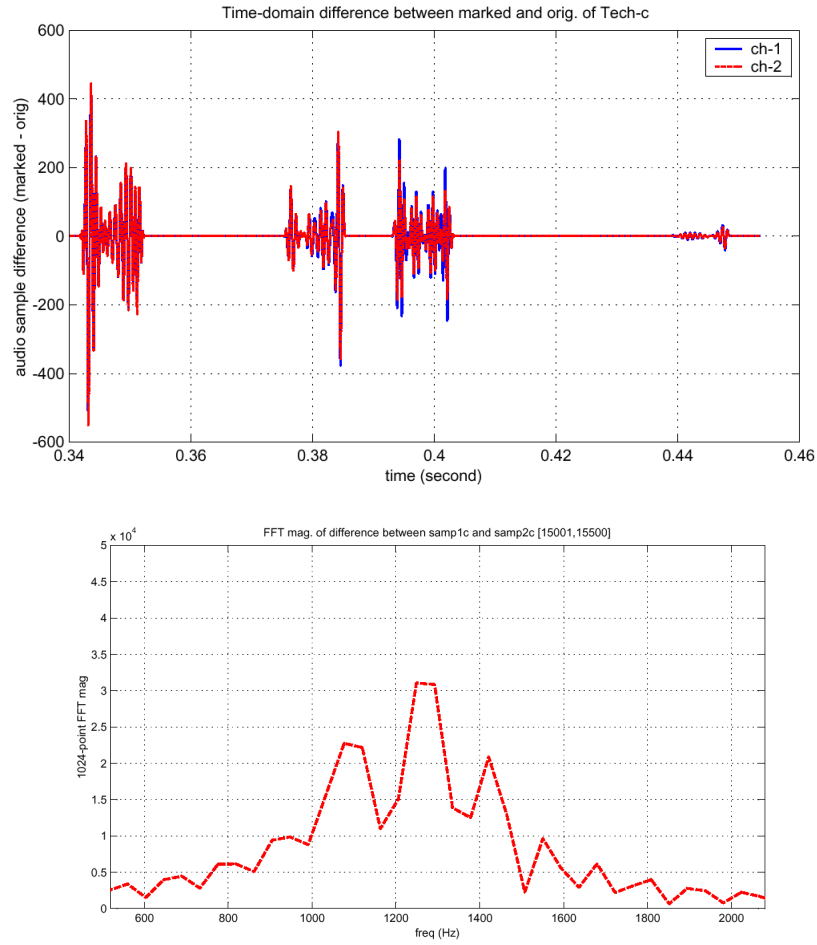


Figure 10.3: Technology-C: (a) the waveform of the difference between sample-1c and sample-2c exhibits tone bursts, and (b) the short-time DFT of one tone burst. The samples observed here occur around 0.34-th second.

in Fig. 10.4, along with several other time-domain jittering/warping that we have encountered during the challenge.

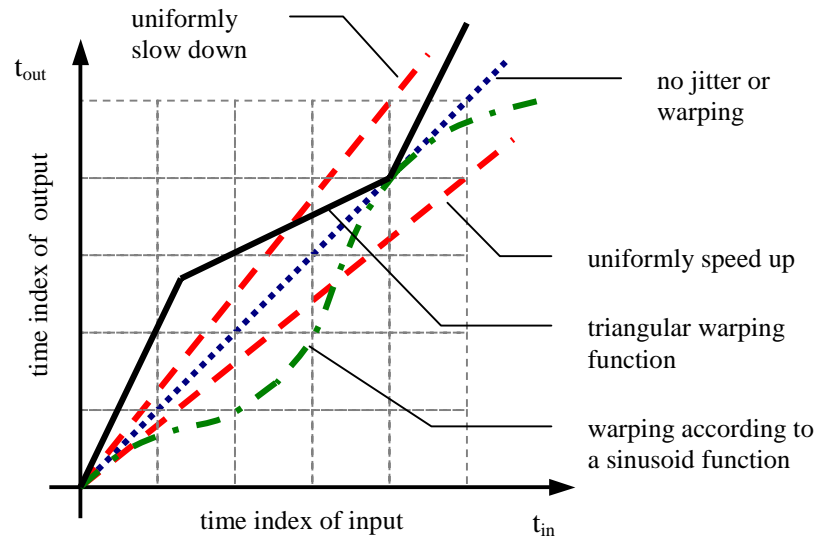


Figure 10.4: Relations between the input and output time index of a few time-domain jittering/warping, including uniform speed-up, uniform slow-down, sinusoid warping, and triangular warping.

One implementation we used is to upsample the audio by M times followed by lowpass filtering and downsampling by N times, giving an overall resampling rate of M/N . The original sampling frequency of F_s is changed to $M/N \cdot F_s$. The resampled audio is then played or stored with the same sampling rate as before, i.e., F_s . The entire process changes the pitch by a fraction of $(N - M)/M$. A precise spectrum interpretation of this can be obtained based on multi-rate signal processing theory [13]. For sampling rate conversion with $M/N > 1$, the spectrum is squeezed along frequency axis by a factor of N/M , leaving the frequency band of $(\frac{N}{M}\pi, \pi]$ with zero; for the case of $0 < M/N < 1$, the frequency band $[0, \frac{M}{N}\pi]$ of the original spectrum is stretched to cover the whole new spectrum, dropping the high frequency band

$(\frac{M}{N}\pi, \pi]$ of the original spectrum. At the end of this rate conversion, the magnitude of the new spectrum is scaled by M/N , the sampling frequency 2π radian per sample corresponds to $\frac{M}{N}F_s$ Hz, and the pitch has not changed. When the signal is played at F_s samples per second, the spectrum with frequency unit of radian per sample is unchanged, but the frequency of 2π radian per sample is now mapped to F_s Hz, effectively changing the pitch by a fraction of $(N - M)/M$. Attack-C1 can also be implemented using commercial audio editing software. For example, the *Effects* \rightarrow *Pitch* menu of *GoldWave v4.19* [148] were used as an alternative way to perform pitch shift attacks (Fig. 10.5).

The ability to detect pitch change varies from individual to individual and depends on whether a reference is available. While most people can discriminate pitch difference as low as 0.6% [102], it is nevertheless rather difficult for a person to identify small pitch changes if he/she has never heard the original before. The standard pitch itself also changed significantly in music history [103, 104]. The pitch of piano's A major, for example, changed steadily from as low as 420Hz in the early 18th century to as high as 457Hz in late 19th century before settling down at the current international standard of 440Hz. Our attack with 3% pitch increase (about a quarter tone) has passed SDMI's strict 2nd round quality testing performed by "golden ears" after the challenge.

As described previously, we observed that the embedding mechanism adds a narrow band signal to the audio at around 1350Hz. Pitch change can be an effective attack because it stretches or squeezes the spectrum, causing misalignment, which in turn reduces the detector response from the popular matched-filter-type detection. One way to enhance the robustness against Attack-C1 is to estimate and undo the stretching, which is likely to be computationally expensive. Another way is to embed

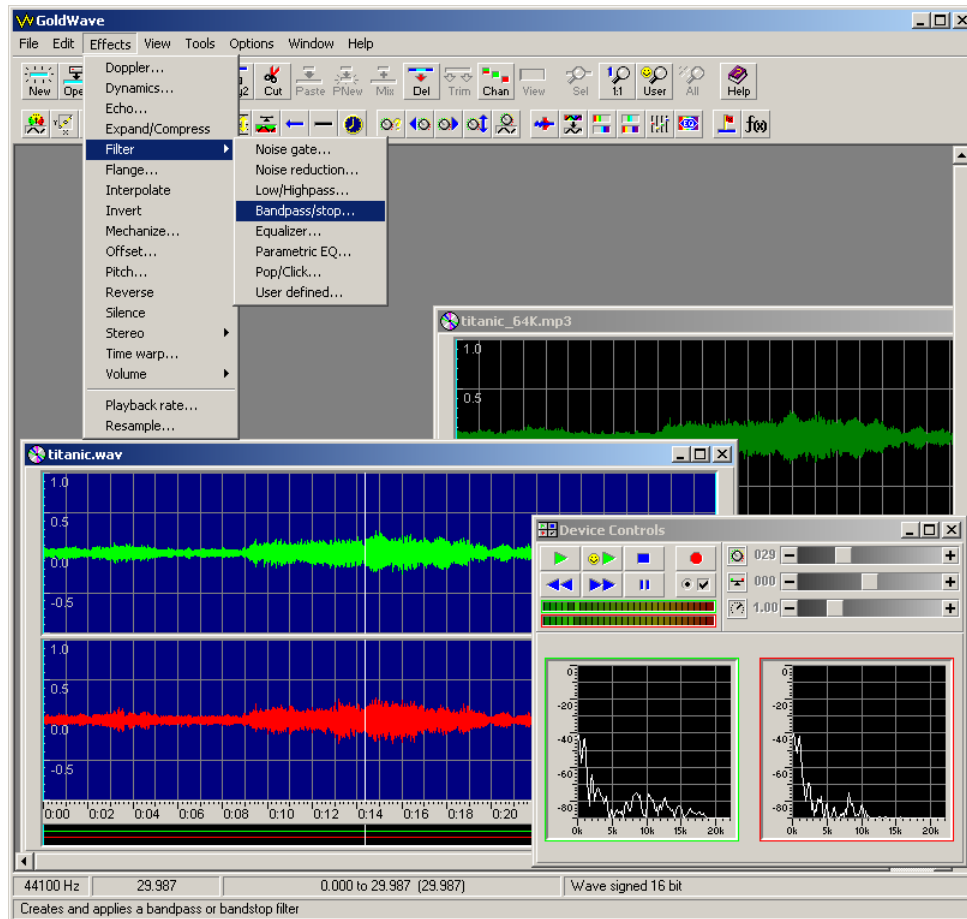


Figure 10.5: Graphics user interface of GoldWave audio editing shareware.

and/or detect watermark in a domain that is resilient to stretching/squeezing.

Attack-C2 Our second attack belongs to Type-2, attempting to jam the frequency band around 1350Hz where it was observed that a narrow band signal had been added by the embedding mechanism. This narrow band watermark signal has some randomness, making jamming difficult. The anti-jamming capability has been seen with the spread spectrum watermark. This commonly used noise-like watermark has good statistical property so that the power of uncorrelated additive noise has to be

large enough to effectively jam the watermark [44]. However, to preserve auditory quality, the noise power has to be kept low. Our successful attack is to apply notch filtering to the audio signal at selected frequencies. The filtering introduces significant changes in magnitude and phase around the notch (shown in Fig. 10.6) [12], effectively damaging the embedded watermark. Specifically, we used the *Effects* \rightarrow *Filters* \rightarrow *Bandpass/stop* menu of the audio editor *GoldWave* to perform notch filtering, with a stop band of 1250-1450Hz and steepness of 5 (i.e., 10th order).

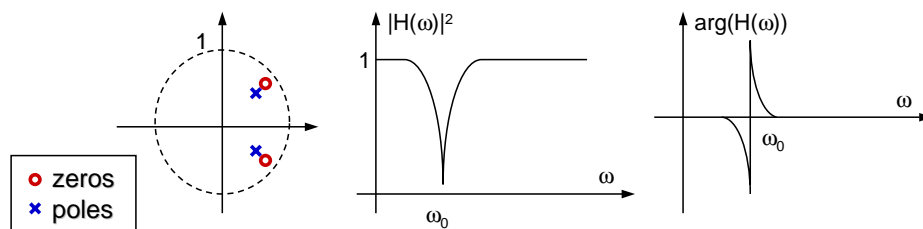


Figure 10.6: A 2nd order notch filter: (from left to right) zero-pole plot and frequency response (magnitude and phase).

Attack-C2 has passed SDMI's 2nd round quality testing performed by "golden ears". For signals with sufficiently rich spectrum, the magnitude and phase changes caused by notch filtering may not be detectable by a person because of frequency masking and other human auditory phenomena. In the next section, we will see that the embedding process of Watermark-B has a step of notch filtering, suggesting that Watermark-B is a potential attack on Watermark-C. It also suggests that the distortion on audio signal imposed by our Attack-C2 is comparable with that by the embedding process of Watermark-B.

10.2.3 Attacks on Watermark A, B & F

Watermark A Our attack on Watermark-A, referred as *copier attack*, is a Type-2 attack. By analyzing the short-time FFT of the samples, we observed regular patterns of phase difference. The observation leads to a time varying model describing the phase difference between sample-1a and sample-2a. Based on the model, our attack “copies” the phase change between sample-1a and sample-2a to sample-3a, aiming at recovering the phase modification done by embedding process. We also introduced some randomness in middle frequency bands during phase manipulation. A variation of this attack incorporating magnitude manipulation was also submitted. Both were confirmed by SDMI oracle as successful.

Watermark B Our attack on Watermark-B is also a Type-2 attack. A spectrum notch is observed around 2800Hz for some parts of the audio and around 3500Hz for some other parts. In addition, the phase difference between original and watermarked audio signals exhibits unique butterfly shape, indicating that notch filtering is involved in embedding. Our attack fills in those notches with random but bounded coefficient values. We also submitted a variation of this attack involving different parameters for notch description. Both were confirmed by SDMI oracle as successful. Interestingly, an embedding technique similar to our observations from Technology-B was found in US Patent 4, 876, 617 “Signal Identification” [112] after the challenge. This once again indicates that relying on the secrecy of the embedding algorithm is not a long-term solution to protecting public watermark system.

Watermark F Our attack on Watermark-F explores the weakness of this watermarking approach under time varying warping in time domain, thus is a Type-1 attack. In particular, we warped the time axis by inserting a periodically varying delay.

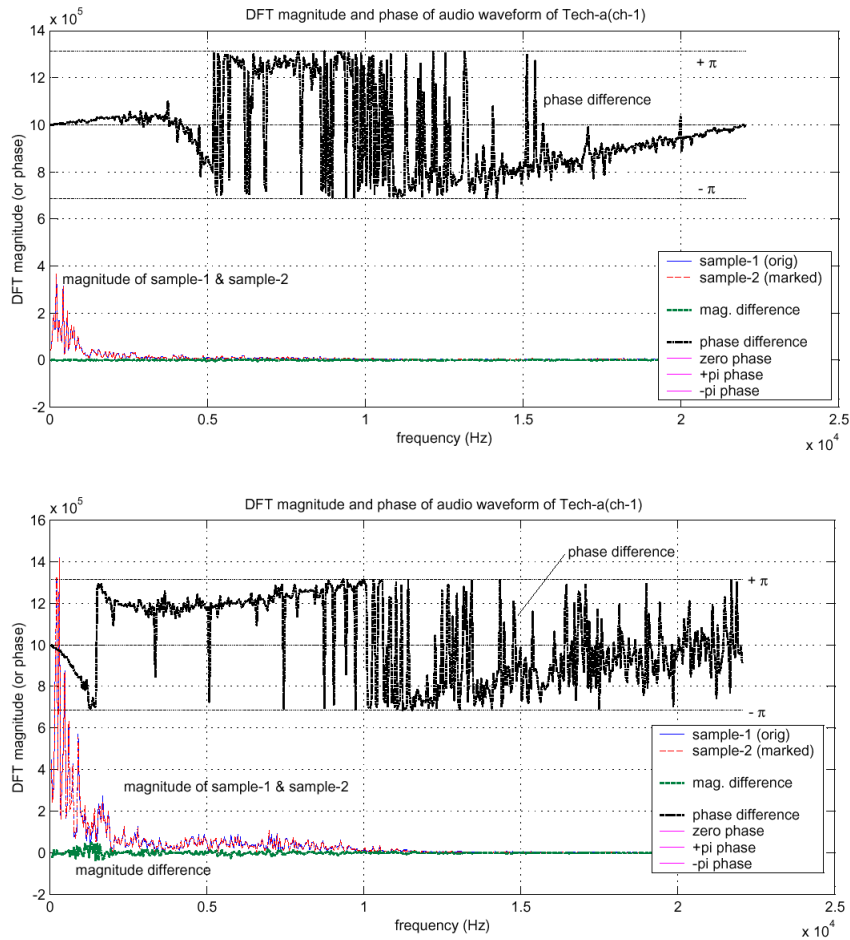


Figure 10.7: Technology-A: FFT magnitude of original and watermarked signals, and phase difference between the two signals for a 1000-sample segment. The two figures are for signals around 3.22-th second and 4.33-th second, respectively.

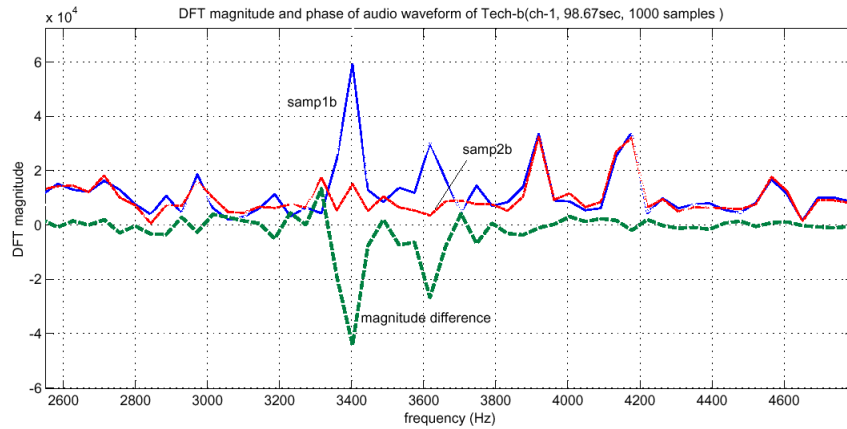


Figure 10.8: Technology-B: FFT magnitudes of sample-1b and sample-2b and their difference for 1000 samples at 98.67 sec.

The delay function comes from our study on Watermark-A, therefore the perceptual quality of our attacked audio is expected to be better than or comparable to that of the audio watermarked by Technology-A. We also submitted variations of this attack involving different warping parameters and different delay function. The warping we performed follows a sinusoid or a triangle function, as illustrated in Fig. 10.4. The attacks were confirmed by SDMI oracle as successful.

Recently, Boeuf and Stern presented their analysis and successful attack on Watermark F in [143]. An autocorrelation analysis was applied to the difference signal between the original and watermarked audio, and a periodicity of 1470 samples ($\frac{1}{30}$ second) was observed. Further study in their report suggested that the difference is a periodic spread spectrum signal with a period of 1470 samples, and the watermark is scaled with different scaling factor for every 147 samples. The scaling factor appears to be a function of the average power of the host audio signal in a local window. The watermark can be detected non-coherently (without using the original audio) by

taking a correlation over a long window to suppress the strong interference from the host signal. This detection strategy has been analyzed in Chapter 3 of this thesis. Having found that the same spread spectrum signal is embedded in both sample-1 and sample-3, Boeuffl *et al.* designed an algorithm to first search for the initial offset from which the watermark starts to be put into sample-3 and to successfully remove the watermark by subtraction. Their work provides a foundation to explain the effectiveness of our blind attack. Without purposely performing registration/synchronization or without embedding in a resilient domain, spread spectrum embedding is vulnerable against jittering².

10.2.4 Summary and Remarks

We presented a general framework for analyzing the robustness and security of audio watermark systems. The framework was demonstrated by our successful attacks in the SDMI public challenge. We pointed out that (1) the weaknesses in the watermarking design are very likely to be explored by an adversary as effective attacks, prompting the need of thorough testing by watermark designers; (2) a large amount of information regarding the embedding mechanism, derived from pairs of original and watermarked signals, can be used to build powerful attacks, prompting the need of obscuring distinct traces between original and watermarked signals. The second point, though not having received much attention in the literature, is crucial for SDMI applications and has a tradeoff with respect to the robustness against other attacks.

Due to various limitations of the challenge including the very short time frame, we adopted practical strategies to increase our chance in finding successful attack(s)

²The counterpart of audio (1-D) warping/jittering attacks in image (2-D) is the geometric distortions. We have discussed the attacks and countermeasures for image watermarks under rotation, scale, and translation in Chapter 9.

and in understanding all four watermark technologies. For example, we did not incorporate sophisticated human auditory system (HAS) models that can further improve the perceptual quality. Instead, we focused on finding attacks that render miss detection by a watermark detector without significantly degrading perceptual quality. As illustrated in Fig. 10.2, instead of starting from highly noisy audio around the point *A*, we look for attacks (such as those around the point *B*) that is as close to high perceptual quality region as possible and in the mean time as far away from detectability threshold as possible. These are crucial start points from which many optimizations, improvement, and fine-tuning can be feasibly made to proceed to the ideal attack region (region IV in Fig. 10.2).

10.3 Proposed Attacks and Analysis on SDMI Fragile Watermarks

We have mentioned earlier that an SDMI system may use both robust and fragile watermarks. In addition to rendering the robust watermarks undetectable, an adversary may forge a fragile watermark to obliterate the access/copy control mechanism. In the example that a policy does not allow lossy compression on audio files, adversaries may first compress an audio file. The lossy compression, which allows the easy exchange of audio files over network, is likely to destroy the fragile watermark but still retain the robust watermark. Before admitting the audio to an SDMI-compliant device, an adversary decompresses the file and forges a fragile watermark. Examining the existence and the content of the robust and fragile watermarks in an audio file, a device draws a false conclusion that the audio has not been compressed and that the user has not violated the access control policy.

More abstractly, the fragile watermark in an SDMI system serves the purpose of tampering detection, which is a major application of fragile watermarks. Issues regarding the designs, the attacks, and the countermeasures of watermark-based authentication have been discussed in Chapter 7, where the basic idea is to keep a reference and to compare with it later. It is desirable to keep the data volume of the reference small so that the overhead in storage or transmission of the entire data is small. The location of the reference does not have to be secret, but the reference must (1) be unambiguous in the sense that two sets of meaningful data are unlikely to have the same signature, and (2) be difficult to tamper without trace. For perceptual source like digital audio, the reference can be combined with the perceptual source in a more seamless way via watermarking. For example, one can embed a prescribed data pattern or some features of the host audio signal into the audio, and later when the authenticity of an audio is in question, one can verify the integrity of these embedded data to decide on the authenticity of the audio signal. The watermark-based authentication relies on either (1) the embedded data, or (2) the fragility and secrecy of the embedding mechanism, or (3) both. Compared with non-embedding approaches that only make use of the first element (e.g., attaching a cryptographic digital signature to the audio), the watermark-based approach may be able to offer additional security if designed properly. A poorly designed watermark algorithm, however, may leave holes for adversaries to forge a valid authentication watermark.

One potential flaw regarding fragile watermark in SDMI-like application is to rely too much on the secrecy of embedding mechanism. In [175], Technology A is taken as an example to demonstrate how the embedding mechanism of a fragile watermark can be explored. Weak echoes have been observed in high frequency bands around 8-16K Hz. The polarity and delay of echoes vary about every 1/50 second, and

they are very likely to be used to encode some authentication information. The data embedded in such high frequency bands are likely to be distorted by lossy compression (such as MP3) and low-pass filtering. If the authentication data (i.e., the data to be embedded) is not wisely chosen, an adversary can explore the inner workings of embedding mechanism and use this knowledge to recover the authentication data after performing unallowed processing/distortion on the audio signal. A trivial choice, for example, to embed the same pattern fragiley for different audio files, could leave holes for adversaries who may repair the authentication data by using the knowledge of the embedding mechanism. Holliman *et al.* discussed a few cases of counterfeiting watermarks in images [139] and pointed out the weaknesses of embedding data that are independent of the host media. If the fragile watermark in an SDMI system were designed to be independent of the host media, it would be vulnerable to forgery attack, implying the perceptual quality of attacked signal could be very good. This is because an attacker does not need to destroy the robust watermark (which could introduce some perceptual distortion, depending on the design and the attack); what he/she needs to do is just to recover the fragile watermark that generally has lower energy and is perceptually transparent.

A countermeasure against forging fragile watermark is to introduce dependency, which has been discussed in Section 9.3. That is, we embed some data, or called “features”, that are derived from the host audio signal. Denoting the features derived from an audio signal S_1 as $d_1 = f(S_1)$, and those derived from an altered signal S_2 as $d_2 = f(S_2)$ (e.g., S_2 could be an MP3 compressed version of S_1), we would like to choose a function $f(\cdot)$ such that d_1 and d_2 are sufficiently different. Encryption and/or cryptographic digest may be used in designing $f(\cdot)$ and the keys associated with $f(\cdot)$ should be kept secret. Readers may notice a potential problem that the

features derived from an audio signal could be different from those derived from its watermarked version, i.e., $f(S_1) \neq f(E(S_1, d_1))$ where $E(\cdot, \cdot)$ is an embedding function. This problem can be easily fixed by embedding the data derived from the i^{th} segment of a watermarked audio in the $(i + 1)^{\text{th}}$ segment of the unwatermarked audio to obtain $(i + 1)^{\text{th}}$ watermarked segment, and so on so forth.

In summary, the fragile watermarks in SDMI-like system should be carefully designed to eliminate weaknesses against counterfeiting attacks and other security holes.

Acknowledgement

The works on attacking four SDMI robust watermark technologies were jointly done with Scott Craver at Princeton University. In particular, the sinusoid warping attack on Technology F was proposed by Craver.