자기 참조 워터마킹을 위한 메시지 코딩 기법

전영균*, 최창렬**, 정제창***
한양대학교 전자통신공학과

Message Coding Technique for Self-reference Watermarking

Yeonggyun Jeon*, Changryoul Choi**, Jechang Jeong***
Department of Electronic Communications Engineering, Hanyang University
*junddaro@ece.hanyang.ac.kr, **deneb@ece.hanyang.ac.kr, **jjeong@ece.hanyang.ac.kr

요약

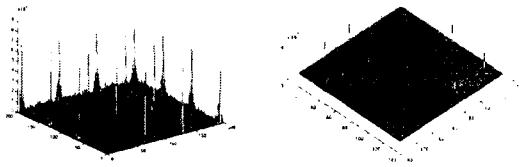
데이터에 삽입되어 있는 워터마크를 제거하려는 공격기법이 점차 발전되면서 여러 가지 워터마킹 알고리즘이 등장하게 된다. 보통 워터마크를 제거하기 위해서는 워터마크 신호의 동기를 제거하는 동기-제거(De-Syncronization) 공격 방법을 이용한다. 이런 공격에 대처하기 위한 워터마킹 방법으로는 세 가지로 분류할 수 있는데, 공격에 영향을 받지 않는 영역(invariant domain)에 워터마크를 삽입하는 방법, 템플릿(template)을 이용한 방법, 자기 참조(self-reference)를 이용한 기법 등이 있다.

본 논문에서는 공간 영역에서의 자기 참조 워터마킹 방법을 소개하고, 이에 적합한 메시지를 삽입하고 검출하는 방법에 대해 제안한다. 삽입하려는 메시지와 Reed-Solomon Code 를 이용하여 만든 부가적인 정보를 QOS (Quasi-orthogonal sequences)방법과 DSSS 를 이용하여 만든 다음 키값을 이용하여 뒤섞으면 워터마킹 블록을 자기 참조 워터마킹 방법으로 영상에 삽입하게 된다. 그 결과 일반적인 필터링 공격에는 95% 이상 검출됐고, JPEG 압축 파라미터 60 에서도 97% 이상 검출되었다.

Ⅰ. 서론

워터마킹(Watermarking)이란 각종 멀티미디어 데이터, 정지영상, 동영상, 오디오, 그래픽 등에 사람의 눈이나 귀로 인지할 수 없는 신호나 내용 등을 삽입하는 방법이다. 요즘 들어 중요성이 점차 증대하고 있는데, 디지털 데이터 처리능력의 발달 및 네트워크의 고속화로 데이터의 불법 복제 및 불법 유포가 용이해져 멀티미디어 데이터와 같은 대용량 데이터도 쉽게 복제가 가능해졌기 때문이다. 데이터에 삽입되어 있는 워터마크를 제거하려는 공격기법이 점차 발전되면서 여러 가지 워터마킹 알고리즘이 등장하게 된다.

보통 워터마크를 제거하기 위해서는 워터마크 신호의 동기(Sync)를 제거하는 방법을 사용하게 된다. 이런 기법을 동기-제거(De-Syncronization) 공격 방법이라 한다. 이런 공격 방법에 대처하기 위해서는 데이터가 어떤 공격을 받았는지 그 정보를 알아내어 공격에 대한 역변환 방법으로 워터마크를 검출하게 된다. 이런 워터마킹 방법으로는 크게 세 가지로 분류할 수 있는데, 공격에 영향을 받지 않는 영역(invariant domain)에 워터마크를 삽입하는 방법, 템플릿(template)을 이용한 방법, 자기 참조(self-reference)를 이용한 기법 등이 있다. 자기 참조 워터마킹이란 주기적으로 워터마크를 삽입하는 방법을 말하는데 ACF (autocorrelation function)를 이용하면 그림 1 과 같이 국부적인 피크를 통해서 어떤 공격을 받았는지 알아낼 수가 있다.



(a) (b)

그림 1. (a) 공격 받지 않았을 때의 Local Peak; (b) 공격 받았을 때의 Local Peak

자기 참조 워터마킹의 특성은 다음과 같다.

- 일반적인 기하학적 공격에 잘 견딘다.
- Reference 를 제거하는 것이 템플릿을 제거하는 것보다 어렵다.
- 국부적인 Warping 공격에는 견디기 어렵다.
- Template 이 사용되지 않았기 때문에 추가적인 노이즈가 생기지 않는다.
- ACF 를 계산하기 위해선 계산량이 많다.

본 논문은 다음과 같이 구성된다. 2 장에서는 Raw Bits Domain 와 Pixel Domain Correlator 에 대해 소개한다. 3, 4 장에서는 공간 영역에서의 자기 참조 워터마킹에 적합한 설계 방법에 대해 설명하고, 5 장에서는 이를 바탕으로 한 메시지 코딩 시스템을 보여주고, 마지막으로 6 장에서는 결론을 맺는다.

Ⅱ. Raw Bits Domain 와 Pixel Domain Correlator

디지털 워터마킹의 일반적인 예는 영상에 메시지 정보를 삽입하는 방식이다. 메시지를 삽입할 때는 대역확산(Spread Spectrum)방식을 이용하여 BER(bit error rate)를 낮추는데, 문제는 확산(spread)된 데이터 하나 하나가 중요한 의미를 가진다는 것이다. 이는 하나의 값에 오류가 생기면 에러가 발생한다는 것을 의미한다. 워터마크를 신뢰 있게 검출하기 위해선 이런 대역확산 방식만으로는 불가능하기 때문에 에러정정기술을 같이 적용하는 게 일반적이다.

M. Kutter[1]는 M-ary 변조방식이 워터마크를 복원하는데 있어 에러 확률을 줄여준다는 것을 보여주었다. 미국 Digimarc 사에 있는 Brett Bradley *et al.* 는 M-ary 변조방식과 에러정정기술과 함께 쓰인 Binary 변조 방식과의 성능 차이

를 소개했다[2]. Brett Bradley *et al.*는 그림 2 와 같은 검출 방식을 소개하고 있다.

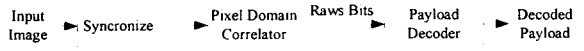


그림 2. Raw Bits Domain Pixel 와 Domain Correlator

여기에서 Pixel Domain 과 Raw Bits Domain 이란 말이 나온다. 위에서 설명한 바와 같이 대역확산 방식처럼 한 비트가 틀리면 에러가 나는 방식이 아닌, 한 비트를 표현하기 여러 개의 픽셀 값을 이용하는 방식을 도입하게 된다. 이를 Pixel Domain 이란 말로 표현하고 있다. 이 다음 과정은 대역확산 방식과 비슷하다. Pixel Domain Correlator 를 거친 값은 마치 대역확산 방식에서 확산된 값들과 비슷하게 생각하면 된다. 이 값들을 Raw Bits 라 표현하고, 이 영역을 Raw Bits Domain 이라 부른다. 그림 2 의 Payload Decoder 는 Binary 변조 혹은 다른 변조 방식과 에러정정 기술이 같이 적용되어 메시지(Payload)를 디코딩하게 된다.

이 방식을 사용하게 되면 여러 가지 이점이 있다. 첫째로는 메시지를 구성하는 각 비트 값들에 대한 신뢰성있는 검출이 대역확산 방식이나 M-ary 변조 방식보다 보장된다. 둘째로는 에러정정기술을 함께 적용할 수 있기 때문에 그 성능을 높일 수 있다. 마지막으로 Pixel Domain Correlator 와 Payload Decoder 부분은 서로 독립적으로 분리될 수 있으므로 각각 성능이 우수한 것들끼리 결합하여 더 좋은 성능을 낼 수 있다는 것이다.

Brett Bradley et al.의 이런 개념을 이용해 메시지 코딩을 위한 시스템을 만들었다.

Ⅲ. Pixel Domain Correlator 설계

Pixel Domain Correlator 설계를 위해서 M-ary 변조 방식을 이용한 실험을 하였다. 하나의 값을 표현하기 위해 몇 개의 픽셀 값을 참조해야만 원하는 검출률과 삽입 가능한 정보량을 만족시킬 것인가를 알아보기 위해서다.

이 실험을 위해선 몇 가지 가정이 필요하다.

- 워터마크의 동기화가 맞았다. (이 가정은 동기화를 위한 추가적인 할당이 필요가 없으므로 메시지 코딩만을 위해 최대의 픽셀 수를 이용할 수 있다.)
- Pixel Domain Correlator 단과 Payload Decoder 단에서는 M-ary 변조 방식을 이용한다.
- 워터마크 블록 사이즈를 128x128 로 정했으므로 참조할 수 있는 픽셀 수는 2^{14} 개이다.
- 삽입하려는 메시지는 16 바이트이다.

첫째 단에서 $32 = 2^5$ 로 변조하면, 둘째 단에서는 $2^{14}/2^5 = 2^9$ 의 픽셀을 참조할 수 있다. 16 바이트를 M-ary 변조를 하려면 $2^9/2^4 = 2^5$ 즉 32-ary 변조를 할 수가 있게 된다. 이런 식의 변조 실험에 대한 조건을 표 1 에 정리하였다.

표기 방법	첫째 단	둘째 단	메시지
F0509_S0504	32-ary 변조	32-ary 변조	16 bytes
F0410_S0604	16-ary 변조	64-ary 변조	16 bytes
F0311_S0704	8-ary 변조	128-ary 변조	16 bytes
F0212_S0804	4-ary 변조	256-ary 변조	16 bytes

표 1. M-ary 변조 실험 표기법

StirMark Benchmark 4.0 을 이용[5]하여 일반적인 필터링 공격과 JPEG 압축 공격을 한 후에 검출률을 정리하였다. 실험 영상은 Fabien A. P. Petitcolas 가 제시한 영상 29 개를 이용[6]했고, 결과에 나타난 수치는 평균값을 의미한다. 평균값이 1 이라는 의미는 29 개 영상 모두에서 검출됐다는 것을 의미한다. 실험 결과는 그림 3 과 같다.

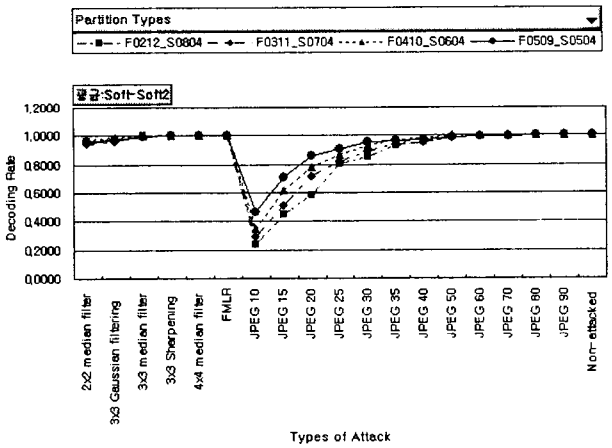


그림 3. M-ary 변조 실험 결과

그림 3 을 보면 전체적으로 첫째 단에서 32-ary 변조를 한 것이 더 좋은 결과를 보인다. M-ary 방식에서의 M 값이 커질수록 성능이 좋은 것과 같은 결과이다. 32-ary 이상의 변조 방식을 사용하면 더 좋은 결과를 거둘 수 있을 지 모르지만, 그렇게 되면 둘째 단에서의 선택 폭에 제한을 두게 되므로 32-ary 변조 방식이 우리가 정한 기준을 가장 만족시킬 수 있으므로 첫째 단은 32-ary 변조 방식을 택하기로 한다.

Ⅳ. Payload Decoder 설계

Binary 변조 방식으로 한 바이트를 표현하려면 8 비트가 소요되며 256 가지의 정보가 있을 수 있다. 우리는 할당된 일정 공간에 더 많은 정보를 넣기를 원하는데, 이는 넣을 수 있는 정보량을 제한한다.

QOS (Quasi-orthogonal sequences)를 이용[3]하면 똑같은 256 가지의 정보를 표현하는데, 5~7 비트로도 표현할 수 있다. 이를 적용하면 더 많은 정보를 넣을 수 있기에 QOS 를 이용한 변조 방식을 도입하기로 했다.

한 가지 주의 해야 할 점은 3 장에서는 워터마크의 동기가 맞았다고 가정했었지만, 워터마크가 삽입된 영상이 동기가 깨지도록 공격 당하면 워터마크를 검출하기가 어렵다는 것이다.

이를 막기 위해 공격을 받더라도 동기를 잘 맞출 수 있도록 2048 개의 픽셀을 동기를 위한 정보를 넣는데 할당한다. 동기를 맞추기에 충분한 길이이다. $2048 = 2^{10} \times 2$ 이므로 1024-ary 변조 방식을 이용할 수 있다.

$2^{14} - 2 \times 2^{10} = 14 \times 2^{10} = 14 \times 2^5 \times 2^5$ 이므로, 첫째 단에서 32-ary 변조를 하고 둘째 단에서 QOS 를 이용하여 변조하는 것이다. QOS 를 이용한 변조 실험 결과는 그림 4 와 같다. 그림에서 QOS (m5b7)이란 7 비트를 표현하기 위해서 $2^7 = 128$ 가지의 정보를 구분할 수 있어야 하는데, QOS 를 이용하여 5 비트로 표현할 수 있도록 변조 했다는 것을 의

미한다. 마찬가지로 QOS (m5b8)는 8 비트를 표현하는데 QOS 를 이용하여 5 비트로 표현할 수 있도록 변조했다는 것이고, QOS (m6b8)는 8 비트를 표현하는데 QOS 를 이용하여 6 비트로 표현할 수 있도록 변조했다는 것을 의미한다. 그림 4 에서 알 수 있듯이 QOS (m5b7)이 가장 성능이 가장 좋게 나왔다. QOS (m5b8)도 좋은 성능을 보이고 있다.

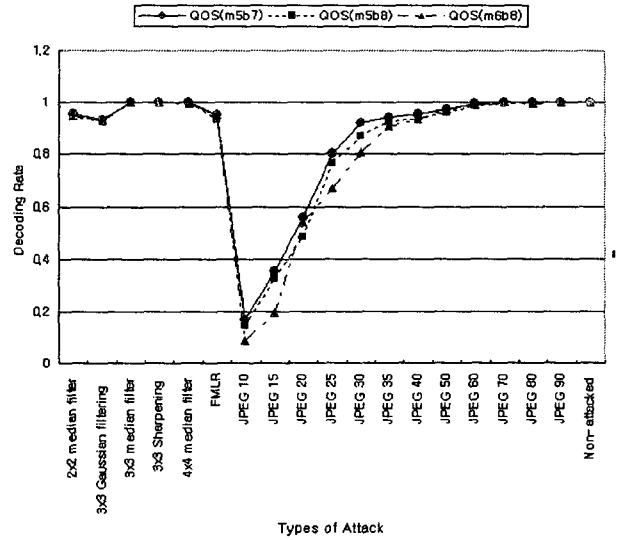


그림 4. QOS 실험 결과 1

이런 QOS 실험 결과와 M-ary 방식 실험 결과와 비교를 해보자. QOS 실험에서는 첫째 단은 32-ary 변조 방식을 사용하고 둘째 단에서 QOS 를 이용한 변조를 했다는 것을 의미한다.

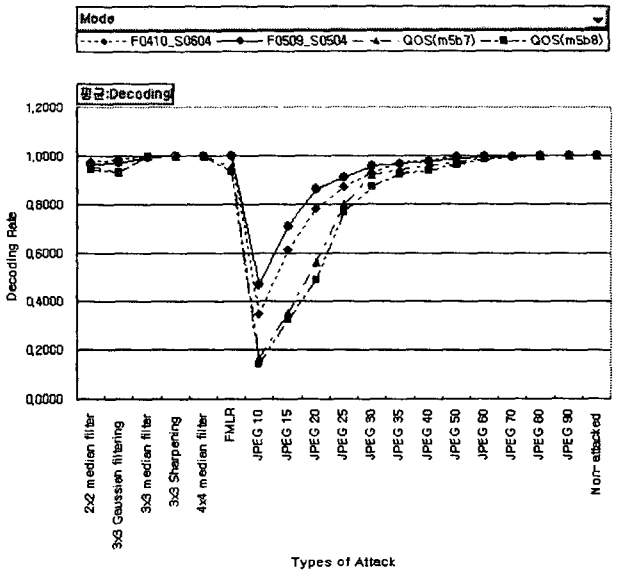


그림 5. QOS 실험 결과와 M-ary 변조 실험 결과 비교

그림 5 에 나온 결과를 보면 두 단 모두 M-ary 방식을 사용했을 때 QOS 를 이용한 변조 방식보다 더 좋은 성능을 보이고 있다. 그러나, 그림을 이해하는데 단순히 수치의 높고 낮음으로 봐서는 안 된다. 삽입할 수 있는 정보량(payload)과 동기화를 얼마나 잘 맞출 수 있는가가 매우 중요한 요소이기 때문이다. 삽입 가능한 정보량을 살펴보면 표 2 와 같다. 두 단 모두 32-ary 변조 방식을 쓸 경우 80 bits 의 정보를 삽입할 수 있다. 그렇지만 표 3 에서와 같이 동기화를 위한 부분이 없기 때문에 워터마크가 공격에 쉽게 깨지기 쉽다.

QOS (m5b7)의 경우는 98 bits, QOS (m5b8)의 경우는 112 bits 의 정보를 삽입 가능하다. 또한 동기화를 위해서 1024-ary 변조 방식을 사용하기 때문에 워터마크가 공격 당하더라도 동기화 부분을 이용하여 동기를 맞추면 워터마크의 검출이 가능해진다. 또한 에러정정기술을 이용하면 같은 정보량(80 bits)를 삽입하면서도 두 단 모두 32-ary 변조 방식을 사용한 것보다 더 좋은 검출률을 거둘 수 있다.

표기 이름	변조 방식		삽입 가능한 정보량
	첫째 단	둘째 단	
F0410_S0604	16-ary	64-ary	96 bits
F0509_S0504	32-ary	32-ary	80 bits
QOS (m5b7)	32-ary	QOS (m5b7)	98 bits
QOS (m5b8)	32-ary	QOS (m5b8)	112 bits

표 2. 변조 방식별 삽입 가능 정보량

표기 이름	변조 방식		동기화 변조 방식
	첫째 단	둘째 단	
F0410_S0604	16-ary	64-ary	X
F0509_S0504	32-ary	32-ary	X
QOS (m5b7)	32-ary	QOS (m5b7)	1024-ary
QOS (m5b8)	32-ary	QOS (m5b8)	1024-ary

표 3. 변조 방식별 동기화를 위한 변조 방식

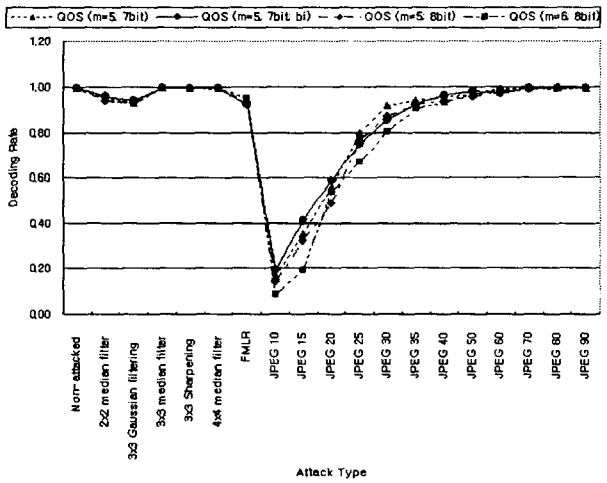


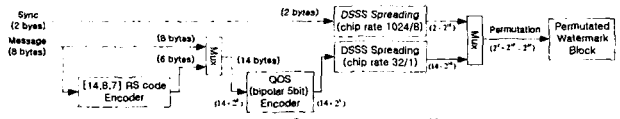
그림 6. QOS 실험 결과 2

검출 성능을 더 높이기 위해서 에러정정기술을 같이 쓰기로 했다. 동기를 맞추는 부분에 할당된 공간을 제외하면 14 바이트의 공간이 남는다. 최대로 넣을 수 있는 메시지량은 14 바이트가 된다. 그러나, 신뢰성 있는 검출을 위해서 [14, 8, 7] Reed-Solomon Code 를 이용[4]하여 에러정정 기술을 사용하면 3 바이트의 에러까지 정정할 수 있도록 했다.

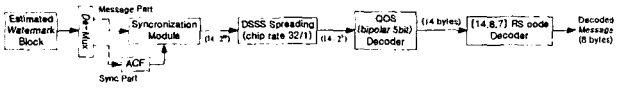
V. System of Message Coding

3, 4 장에서 설계한 모듈을 전체적으로 구성하면 그림 7 과 같다. 메시지 코딩의 부호화기에서는 먼저 워터마크 블록을 형성하는데, 들어가는 정보는 동기화를 위한 부분과 정보 부분으로 나눠진다. 동기화 부분은 DSSS (Direct

Sequence Spread Spectrum)를 이용하여 만들고, 정보 부분에 해당하는 데이터는 메시지와 Reed-Solomon Code 를 이용하여 만든 부가적인 정보를 QOS (Quasi-orthogonal sequences)방법과 DSSS 를 이용하여 만든다. 이렇게 만든 워터마크 시퀀스를 키값을 이용하여 뒤섞으면 워터마킹 블록이 만들어진다. 이렇게 만든 워터마킹 블록을 전체 영상 크기만큼 반복하여 전체 영상 크기의 워터마크 데이터를 만들어 삽입강도를 조절하여 워터마크를 삽입하였다. 메시지 코딩의 복호화기에서는 워터마크를 예측하고, ACF 를 이용하여 동기 정보를 추출한다. 이를 이용하여 메시지 부분에 해당되는 정보들의 동기를 맞춘다. DSSS 와 QOS 를 이용하여 복호화하고, RS Code Decoder 를 이용하여 에러를 정정하게 된다.



(a) Encoder of Message Coding



(b) Decoder of Message Coding

그림 7. System of Message Coding

VI. 결론

본 논문에서는 자기 참조 워터마킹에 적합한 메시지 코딩 방법에 대해 제안하였다. 워터마크의 검출률 기준은 응용 분야에 따라 다르지만, 제한된 환경에서 최적의 워터마킹 시스템을 구축하는 예를 보였다. 워터마크 블록 사이즈를 128x128 로 했을 때 14 bytes 의 정보량을 삽입할 수가 있었다. 에러정정을 위한 부가적인 6 bytes 를 제외하면 순수한 메시지는 8 bytes 가 가능하다.

본 논문에서는 워터마크의 검출률을 높이기 위해 두 가지 방법을 사용했다. 첫째는 워터마크의 동기화를 위해 1024-ary 변조 2048 개의 픽셀을 참조하도록 하였다. 이로써 워터마크가 삽입된 영상이 공격을 받더라도 ACF (autocorrelation function)을 이용하여 동기를 찾을 수가 있었다. 둘째로 각각의 정보 한 비트가 높은 신뢰도를 갖게 하기 위해 Pixel Domain 과 Raw Bits Domain 개념을 도입하여 메시지를 두 단에 걸쳐 각기 최적의 모듈을 구성하였다. 그 결과 공격을 받았을 때 일반적인 대역확산 방식보다 더 좋은 성능을 보장 받을 수 있었다.

VII. 참고문헌

[1] M. Kutter, "Performance Improvements of Spread Spectrum based Image Watermarking Schemes Through M-ary Modulation," *Preliminary Proceedigs of the third International Information hiding Workshop*, pp. 245-260, Dresden, 1999.

[2] Bret Bradley and Hugh Brunk, "Comparative performance of watermarking schemes using Mary modulation with binary schemes employing error correction coding," *Proceedings of SPIE*, Vol.4314, 2001

[3] K. Yang, Y. Kim, and V. Kumar, "Quasi-Orthogonal Sequences for Code-Division Multiple-Access Systems," *IEEE Trans. Information Theory*, vol. 46. pp. 982-993, May 2000.

[4] GMD FOKUS - R & D - Competence Centers - MOBIS – FEC http://www.fokus.gmd.de/research/cc/mobis/products/fec_old/content.html

[5] StirMark Benchmark 4.0 http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark/

[6] Watermarking Image Database http://www.cl.cam.ac.uk/~fapp2/watermarking/image_database/index.html