증명서의 온라인 발급을 위한 텍스트 임베딩기법에 관한 연구

최기철* 최종욱**

A study of text embedding technique for issuing digital Certificate

Jizhe Cui, Jong-UK Choi

요약

최근 전자상거래가 활성화되면서, 거래 인증서와 같은 온라인 증명서가 광범위하게 사용되고 있다. 그리고, 증명서의 위/변조기술이 발전함에 따라서 온라인 거래에 사용되는 증명서의 인증과 위조/변조를 방지하는 기술이 필요하게 되었다. 본 연구는 증명서의 인증에 필요한 기술로서, 메시지 인증함수가 가지는 성질을 포함하고 있다. 본 연구에서 개발한 알고리즘은 증명서에 포함된 텍스트문서가 위조/변조 되었을 경우 그 변동 상황을 알아내며, 부정적으로 위조/변조된 부분을 검출하며, 변동상황 검출과 함께 원 증명서의 문서를 복원할 수 있는 기술이다. 만일 이 증명서에 대하여 변동이 진행된 흔적이 발견될 경우, 증명서를 인증하지 않으며, 삽입한 텍스트 데이터를 추출하고 변동을 확인하는 것과 함께 필요한 정보를 복원한다. 본 논문의 시험결과에 근거하면 256×256 BMP file Format 이미지에 3만2천자 정도의 텍스트문서를 삽입할 수 있었다.

Key words : 인증서, 제어문자, Text embedding, Mapping Table, Seed Number

1. 서론

온라인 거래에서는 발급자 및 사용자의 인증서, 온라인 결재 증명서와 같은 증명서류의 교환이 필수적이다. 온라인 거래를 위하여 특정형식의 증명서를 주고 받는 경우에는 문서의 위/변조가 비일비재하게 발생되므로 저작물의 원본과 복사본, 위조/변조 본을 식별할 수 있는 기술적 장치가 필요하다.

본 연구에서는 전자상거래에서 유용하게 활용할 수 있는 증명서의 발급과 확인을 위해서 다음 기능을 갖는 기술을 개발하였다.

1) 온라인 증명서의 위조/변조를 가려낼 수 있는 기능
2) 온라인 증명서의 위조/변조를 가려낼 수

* 상명대학교 대학원 컴퓨터 과학과(jzcui@markany.co.kr)
** 상명대학교 정보통신대학원 교수(juchoi@markany.co.kr)
본 연구는 정보통신부 대학기초연구(C1-98-0673)에 의해 지원되었음.

있는 기능

3) 온라인 증명서가 위조/변조 되었을 경우, 위조/변조된 부분을 가려내는 기능
4) 온라인 증명서가 위조/변조 되었을 경우 위조/변조된 내용을 복구할 수 있는 기능
5) 온라인 증명서에 대한 서명기능과 인증기능
6) 거래 상대방에 대한 신원확인 기능

2. 증명서특성 및 삽입공간

2.1 증명서의 특성

온라인 거래에 사용되는 증명서는 아래와 같은 세가지 특징이 있다.

첫째, 발행되는 증명서에는 법인, 저작권자, 혹은 발행자임을 확인하기 위하여 이미지형태의 사진이나 인감, 회사의 로고, 등이 삽입된다. 법인, 저작권자, 혹은 발행자의 사진이나, 인감, 회사의 로고의 사용은 증명서의 신뢰성을 높여 주었다.[그림2-1]

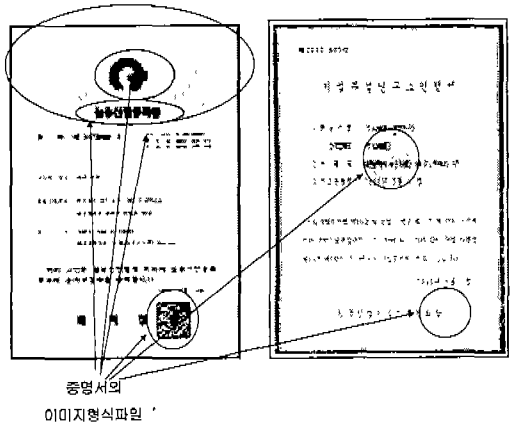

그림 2-1 증명서와 이미지형태의 문서

그림2-1는 증명서에 흔히 사용되는 이미지 형태의 문서를 분석하여 보여준 것이다. 이러한 온라인으로 발급되는 증명서 및 공증서 등이 가지고 있는 특징은 워터마크를 삽입할 공간을 찾는데 큰 의미가 있다. 일본의 Munetoshi IWAKIRI, Yoshio MURAKAMI 등은 특수한 인감 이미지의 특성을 이용하여 문서의 변동여부를 찾아내고 만약 변화되었다면 복원할 수 있는 기능을 추가한 기법을 개발하였다[1].

그림 2-2 증명서 내의 이미지형태의 문서

둘째, 발급되는 증명서는 발급자를 승인하거나 저작권자의 권리를 보호하기 위하여 지정된 특별용지를 사용하는 경우이다. 온라인으로 배포하는 텍스트문서의 형식이나 배포하는 회사가 다름에 따라서 사용하는 용지가 틀리지만 인증을 위하여 일반 용지를 쓰는 경우는 적다.

셋째, 상술한 특별용지를 사용하지 않거나, 이미지 형태의 파일이 포함되지 않는 증명서는 아주 적다.

본 연구에서는 증명서가 가지는 이러한 특성을 이용하여 증명서에 포함되는 이미지를 삽입공간으로 하여, 증명서의 보호를 위한 텍스트문서를 이미지에 삽입한다.

2.2 이미지 구성과 삽입구간

이미지는 크게 매트릭스형 이미지와 벡터형 이미지로 나뉜다. 본 논문에서는 증명서에 사용되는 이미지 중, 98% 이상을 차지하고 있는 매트릭스형 이미지에 대해 분석하고 삽입구간을

탐색한다.

매트릭스형 이미지는 단색의 명암 변화 이미지의 경우와, 단색이 아닌 컬러 이미지의 경우가 있다. 컬러의 천연색을 다루기 위하여 빨강, 초록, 파랑의 3원색으로 분해한 3장의 이미지를 작성하고 이것들을 각각 이미지 처리하여 출력할 때 합성해서 컬러 이미지를 구성한다. 일반적으로 칼라이미지는 RGB모델을 사용하지만 때때로 다른 칼라모델을 사용하는 경우도 있다.

본 논문에서는 증명서에 포함되는 이미지의 공간영역 상에 텍스트형식의 워터마크를 삽입한다. 흑백이미지의 경우 흑백이미지 전체영역을 삽입공간으로 적용한다. 칼라 이미지의 경우 R-영역, G-영역, B-영역을 각각 삽입공간으로 적용할 수 있는데 사용자의 의도에 따라 임의로 선택하거나 모두 삽입공간으로 적용할 수 있다. 증명서에 이미지형태의 파일이 존재하지 않을 경우, 텍스트 문서를 보이지 않는 이미지형태로 생성하여 증명서에 삽입하는 것으로 텍스트 문서의 삽입공간을 해결했다.

3. 삽입 알고리즘

3.1 텍스트 문서 생성

기존의 ASCII code를 변형시켜서 텍스트문서의 생성에 이용한다.

ASCII는 94개의 출력용 문자와 34개의 여러 종류의 제어용으로만 사용되는 출력되지 않는 제어문자를 가지고 있다. 제어문자는 데이터의 목적지 제어를 하고 출력되어질 텍스트를 배열하는 데 사용한다. 제어문자에는 규정자(format effectors), 정보구분자(information separators)와 통신제어문자(communication-control character)가 있다. 규정자는 출력의 규정(layout)을 제어하며, backspace(BS), horizontal tabulation(HT), carriage return(CR) 등의 친숙한 타이프용 제어들이 있다. 정보구분자는 데이터들을 문단 혹은 페이지 등으로 나누는데 사용되며, record separator(RS), file separator(FS) 등이 있다. 통신제어문자는 STX(start of text), ETX(end of Text) 등 전화선을 통해 텍스트 메시지가 전달되는 규정을 만들 때 쓰인다.

삽입할 텍스트 문서의 구현에는 34개의 제어문자를 사용할 필요가 없다. 삽입할 정보의 구현에는 26개 대문자, 26개 소문자, 10개 숫자와 32개 특수문자면 충족하다. 그 중 32개의 특수문자도 몇 개의 특정문자를 제외하면 거의 사용이 되지 않고 있다. 따라서 실지 필요로 하는 문자는 62개~ 94개이다. 여기서 대소문자를 구분하지 않는다면 36~68개의 문자범위를 가진다. 본 논문에서 워터마크를 제조할 때 이러한 36~68범위내의 문자를 사용한다. 본 논문에서는 사용하는 문자를 64개로 제한하였다.

삽입할 텍스트문서의 생성에는 다음과 같은 열 가지 규칙을 적용했다.

1) 원 텍스트문서의 대문자는 모두 소문자로 변환시킨다.
2) 원 텍스트문서의 "{" , "}"는 모두 "("로 변환시킨다.
3) 원 텍스트문서의 "[" , "]"는 모두 "("로 변환시킨다.
4) 원 텍스트문서의 "(" , ")"는 모두 "("로 변환시킨다.
5) 원 텍스트문서의 "-", "—", "~"는 모두 "-"로 변환시킨다.
6) 원 텍스트문서의 " ' " 와 " ` "는 모두 " ' "로 변환시킨다.
7) 원 텍스트문서의 숫자와 소문자는 그대로 적용한다.

8) 원 텍스트문서의 ASCII Code LF와 CTR은 그대로 적용한다.

9) 원 텍스트문서의 나머지 문자들은 모두 Space로 대치한다.

10) 원 텍스트문서의 글꼴, 글꼴 스타일, 문자 크기, 문자 간격, 텍스트 효과 등은 모두 무시한다.

상술한 삽입할 텍스트 문서 생성 시 사용되는 문자를

$$X_i \quad i = 1, 2, \cdots, 63$$

라 하며 사용되는 문자의 개수를 I 라고 한다. 양자화 이론에 의하여

$$\log_2 I \leq 6$$

임을 알 수 있다. 따라서 6비트의 길이를 가지는 바이너리 코드로 원본 텍스트 문서를 나타낼 수 있다. 원 텍스트 문서를 상기 방법으로 변환시킨 후 삽입을 위하여 6bit Binary Code로 비트열화 한다.

표 3-1 Mapping table Index

CHAR	INDEX	CHAR	INDEX	CHAR	INDEX	CHAR	INDEX
a	0	q	16	6	32	`	48
b	1	r	17	7	33	-	49
c	2	s	18	8	34	.	50
d	3	t	19	9	35	/	51
e	4	u	20	SP	36	:	52
f	5	v	21	!	37	;	53
g	6	w	22	"	38	<	54
h	7	x	23	#	39	=	55
i	8	y	24	$	40	>	56
j	9	z	25	%	41	?	57
k	10	0	26	&	42	@	58
l	11	1	27	'	43	[	59
m	12	2	28	(	44	₩	60
n	13	3	29	)	45	^	61
o	14	4	30	*	46	{	62
p	15	5	31	+	47		63

Step2: Mapping Table에 의해 참조한 INDEX를 6비트의 이진수로 바꾸어준다.

Step3: 각 6비트 이진수로 바꾸어서 얻어진 비트 열을 순서대로 열거하여 원하는 비트 열을 얻는다. 비트 열을 $B(j)$라고 정의한다.

$B(j)$는 삽입될 비트 열이다.

각 문자가 6비트의 문자열 형태를 취하므로 X_i의 i와 $B(j)$의 j는 $j = 6i$ 와 같은 관계가 있다.

아래에 그림3-2는 워터마크 문서의 비트열화하는 과정을 표현한 것이다.

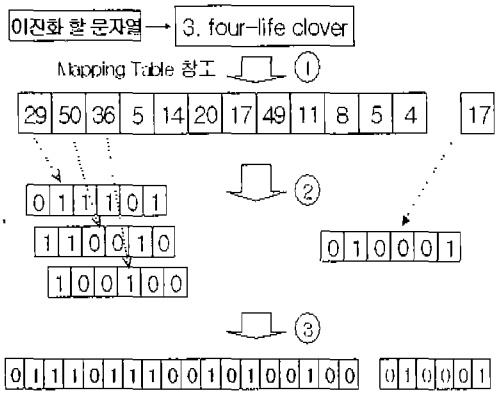

그림 3-1 비트열화 과정

3.2 삽입할 텍스트 문서의 비트열화

삽입하려는 원 텍스트 문서를 변환 시킨 후 사용되는 문자 X_i의 개수는 64개이다. 이러한 64개의 문자를 증명서에 포함되는 이미지에 삽입한다. 삽입 시, 비트 열 형태를 취한다. 따라서 이진화가 필요하다. 변환 후 문서의 이진화는 다음과 같은 절차로 진행한다.

Step1: 삽입하려는 문자 X_i를 아래의 Mapping Table에 근거하여 0~63 사이의 숫자로 Mapping시킨다. 즉, 아래의 표1의 Mapping Table의 INDEX와 같다.

3.3 삽입 알고리즘

비트 열 $B(j)$를 다음과 같은 절차를 거쳐 이미지에 삽입한다.

삽입과정

Step1: 증명서류에서 포함될 이미지 포맷을 가지는 데이터를 취한다. 개인이나 발급자의 인

감이미지, 사용자의 증명사진, 발행회사의 로고 이미지 등이 포함된다

Step2: 증명서류의 텍스트 문서를 분석하여 삽입할 데이터를 작성하며 텍스트 문서 생성 과정을 거쳐서 64개의 코드로 표시하여 삽입할 텍스트 문서를 생성한다.[3.1의 텍스트문서 생성 과정]

Step3: 삽입할 텍스트 문서를 생성한 후 비트 열화 과정을 거쳐서 $B(j)$를 얻는다. $B(j)$가 삽입되는 정보이다[3.2의 삽입할 텍스트 문서의 비트열화].

Step4: 증명서에 포함된 이미지에 RSI(Raster Scan Image)기법을 적용하여 1차원데이터로 변환한다. RSI는 이미지의 왼쪽 위를 시점으로 하여, 최상위 행(열)부터 차례로 하행(다음 열)인 이미지의 Pixel 값을 1차원적으로 재 배열시켜, 1차원 신호를 작성하는 방법이다. 증명서류의 이미지가 단색일 경우 RSI과정을 거쳐서 2차원 데이터를 1차원 데이터로 변환하고 컬러 이미지일 경우는 Red, Green, Blue 성분으로 분해한 다음 각 각 RSI과정을 거쳐서 1차원데이터로 변환한 다음 순서대로 결합하여 1차원 데이터로 변환한다.

이미지 데이터를 1차원으로 변화시킨 결과를 $image(i)$라 하고 이미지의 사이즈를 $N \times N$이라고 하면 i의 범위는 단색일 경우

$$0 \le i \le N \times N \quad (1)$$

이고 칼라 이미지일 경우

$$0 \le i \le N \times N \times 3 \quad (2)$$

이다. $image(i)$가 취하는 값의 범위는

$$0 \le image(i) \le 2^m - 1$$

이다. 여기서 m은 이미지의 양자화 비트수이다.

Step5: Max($image(i)$)을 Max($image(i)$)-1로 바꾸어준다.

이는 텍스트 문서를 삽입한 후 픽셀의 최대값이 $0 \sim 2^m - 1$범위를 벗어나지 않게 하기 위한 과정이다.

Step6: 증명서류의 발급자 혹은 증명서류의 소유자가 자신만의 정보임을 확인할 수 있도록 하기 위하여 십진수 8자리 정수로 된 키(Seed Number)를 가지도록 한다. 이 키에 의하여 워터마크가 삽입되는 위치가 정해지게 된다. 여기서 키 값을 Key_{SN}라고 한다.

Step7: Step6에서 얻은 키 값 Key_{SN}에 의하여 이미지에서 워터마크를 삽입할 위치를 정한다.

Step8: 워터마크 삽입원리에 의하여 비트 열을 이미지에 삽입한다. Step7에서 정한 위치 즉, Key_{SN}+1번째 픽셀의 수치로부터 시작하여 삽입을 진행한다.

$B(i)$의 값과 $Key_{SN}+1$번째 픽셀 값 $image(key_{SN}+i)$를 비교하여 만약,

$$image(key_{SN}+i) \equiv B(i) \mod 2$$

이면 이미지의 픽셀 값 $image(key_{SN}+i)$은 변화시키지 않는다. 만약,

$$image(key_{SN}+i) \; != B(i) \mod 2$$

이면 $image(key_{SN}+i)$의 값을 $image(key_{SN}+i)+1$로 바꾸어준다.

이러한 과정을 반복하여, 이진 비트 열 $B(i)$의 값에 근거하여 이미지의 픽셀 값을 변화시켜서 결과적으로

$$image(key_{SN}+i) \equiv B(i) \mod 2$$

가 되도록 한다. 이러한 기법을 적용하여 비트 열 $B(i)$의 모든 정보를 이미지의 픽셀 값 $image(i)$에 표현한다.

그림4 는 이러한 삽입 알고리즘의 진행 절차를 보여준다.

삽입 알고리즘

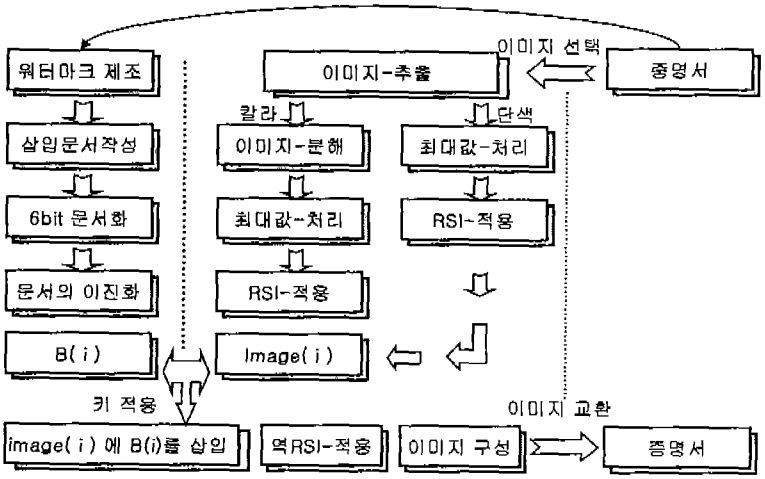

그림 3-2 삽입 알고리즘

삽입 알고리즘에서 볼 수 있는 바와 같이 증명서에 의거하여 이미지를 선택/추출하고, 삽입할 워터마크 문서를 제조한다. 증명서에서 선택한 이미지에 증명서에 의하여 작성한 워터마크 문서정보를 삽입한 후 원 이미지와 구별할 수 없는 형태로 작성하여 증명서의 원 이미지의 위치에 삽입하고 원 이미지는 사용하지 않는다. 이러한 과정을 거쳐서 텍스트 문서의 삽입을 실현하여 증명서의 인증을 할 수 있다.

3.4 키 값의 범위와 역할

가. 키 값의 범위

키는 증명서 소유자 혹은 증명서 발급자가 임의로 정한 1~8자리의 십진수에 의하여 정해진다. 이 키를 그대로 사용하는 것은 아니다. 수학적인 연산을 거쳐서 키의 값이 지정한 범위에 들어가도록 한다. 사용되는 키는 아래와 같은 두 가지 조건의 제약을 받는다.

제약조건1, 키는 증명서의 이미지 픽셀 개수 즉, 이미지의 사이즈와 밀접한 관계가 있으며 이미지 사이즈의 영향을 받는다. 이미지 사이즈가 삽입할 수 있는 비트열의 범위가 된다. 여기서 $image(i)$의 정의역 i가 가지는 최대치를 $Max(image,i)$라고 한다.

제약조건2, 삽입할 이진 비트열의 길이 즉, $B(i)$에서 i가 가지는 값의 최대치의 영향을 받는다. 여기에 의해 키의 길이가 결정되게 된다. 여기서, $B(i)$에서의 i가 가지는 최대치를 $Max(B,i)$라고 한다.

키 Key_{SN}가 가질 수 있는 값의 범위는 D라고 한다. D의 값은 비트 열의 최대치 $Max(B,i)$와 이미지 사이즈에 얻은 삽입할 수 있는 비트열의 최대치 $Max(image,i)$와 다음과 같은 관계가 있다.

$$D=\{\forall key_{SN} \mid 0 \le key_{SN} \le Max(image,i)-Max(B,i)\}$$

만약, 입력한 Key_{SN}의 값이 $Max(image,i)-Max(B,i)$값보다 크면 키 값을 아래의 연산에 근거하여 변화시켜서 사용한다.

입력한 Key_{SN}의 값이,

$Max(image,i)-Max(B,i)$값보다 작으면 그대로 사용한다.

즉, 실제 사용되는 값을 key_{SN1}라고 하면,

$key_{SN1}=$

$Key_{SN} \mod Max(image,i) - Max(B,i)$

이다.

나. 키의 역할

증명서 발급자, 혹은 사용자만 알고 있는 키는 증명서의 발급자 혹은 증명서의 소유자가 자신만의 정보임을 확인할 수 있도록 하기 위하여 사용된다. 키 값을 이용하여 어떤 사람이 텍스트 문서를 위조한 후, 증명서의 이미지에 삽입된 문서도 본 알고리즘을 이용하여 확인하려고 하는 경우를 막을 수 있다.

키 값을 이용하여 삽입 구간을 정했기에 만약 키 값의 보안을 유지하면 워터마크 자체의 안정성도 보장되게 된다

4. 추출 알고리즘 및 절차

Step1: 증명서에 포함되는 개인이나 발급자의 인감이미지, 사용자의 증명사진, 또는 발행회사의 로고 이미지와 같은 워터마크가 삽입된 이미지를 뽑아낸다.

Step2: 뽑아낸 이미지가 단색이미지인가, 칼라 이미지인가를 구분하고 칼라 이미지인 경우 Red, Green, Blue 성분으로 분해한다.

Step3: RSI기법을 적용하여 이미지 픽셀 값을 1차원 데이터 열로 배열한다.

Step4: 사용자/발행자가 가지고 있는 키 Key_{SN} 의하여 삽입이 시작된 위치를 찾아내고 $image(Key_{SN}+1)$ 에서 시작하여

$$image(key_{SN}+i) \equiv B(i) \mod 2$$

를 만족시키는 비트열 $B(i)$를 구성한다.

Step5: 구성한 비트 열 $B(i)$를 6비트씩 묶어서 0~63 사이의 십진수로 바꾸어준다.

Step6: 구성한 십진수 열을 Mapping Table에 근거하여 64개 문자를 가진 문서로 전환한다.

Step7: 얻어진 결과는 우리가 복원하려는 텍스트 문서가 된다.

삽입된 텍스트 문서를 복원하는 과정은 삽입 과정의 역 과정이다. 삽입 시, Mapping Table을 이용하여 사용되는 문자의 개수를 64개로 제한하였기에 복원한 텍스트문서는 Mapping Table에 있는 64개 문자로 구성되게 된다.

5. 성능평가 및 결과

5.1 삽입정보의 양

텍스트 문서로 삽입하는 정보의 양은 증명서에 포함되어 있는 이미지의 사이즈와 그 이미지의 양자화 수준과 직접적인 관계가 있다. 아래의 표는 양자화 비트 수와 그 이미지 사이즈에 따른 워터마크 삽입량의 계산 결과이다.

표 5-1 텍스트 문서의 삽입 양

사이즈	양자화값	삽입 양
256 by 256	칼라(24bit)	32,768자
256 by 256	단색(8bit)	10,922자
153 by 134	칼라(24bit)	10,251자
153 by 134	단색(8bit)	3,417자

삽입한 정보의 양에서 볼 수 있는 바와 같이 증명서에 이미지가 포함되어있을 경우에는 그 내용의 제한을 거의 받지 않는다. 보통 증명서는 500자~10,000자 정도의 내용을 가진다. 따라서 삽입할 문서의 사이즈의 제한을 거의 받지 않는다.

5.2 이미지 변화

텍스트 문서를 이미지에 삽입하는 과정은 이미지에 노이즈를 삽입하는 과정이다. 그림1, 그림2, 그림3은 인감이미지, 단색이미지, 칼라이미지에 텍스트 문서를 삽입하고, 삽입 전 이미지와 삽입 후의 이미지의 변화를 비교 분석해본 결과이다. 시각적인 확인은 불가능했지만, 히스토그램의 분석으로 그 변화를 확인 할 수 있었다.

그림 5-1 단색이미지의 삽입 전/후

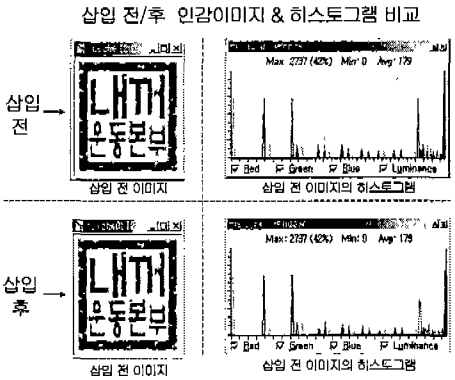

그림 5-2 인감이미지의 삽입 전/후

5.3 이미지처리에 대한 영향

이미지처리에는 많은 종류가 있다. 본 연구에서는 워터마크가 삽입된 이미지에 대해 여러 가지 이미지처리를 진행하였다.

텍스트문서가 삽입된 이미지에 JPEG과 같은 주파수 공간에서의 압축을 진행할 경우 삽입된 텍스트문서가 사라졌다. 유사한 방법인 특징추출필터, 평활화필터, 미분필터를 적용했을 경우에도 삽입된 텍스트문서를 복원할 수 없었다. 다른 이미지처리기법으로 히스토그램의 균일화(Histogram equalization), 이미지 예리화(Image sharpening), 이미지 크로핑(Image cropping)을 적용해 보았다. 그 결과 여전히 삽입된 텍스트 문서를 복원할 수 없었다.

회전을 비롯한 일부 기하학적인 이미지처리를 제외한 거의 모든 이미지처리기법에 의하여 삽입된 텍스트문서를 추출할 수 없었다.

취약한 워터마크(Fragile Watermark)의 성질이 그대로 표현되었다. 표 4-5는 이미지처리 분석결과이다. (X-추출불가능, O-추출가능)

표 5-2 이미지처리기법 & 추출가능성분석

이미지 처리기법	칼라	단색	인감
JPEG압축	X	X	X
cropping	X	X	X
양자화	X	X	X
Sharpening	X	X	X
필터링	X	X	X
회전변화	O	O	O

본 연구에서 구상한 취약한 워터마크(Fragile Watermark)의 성질이 그대로 나타났다.

5.3 결과 및 향후 연구방향

본 논문에서는 증명서와 같은 전자문서의 변경, 위조를 방지하는 기법을 제안했다. 증명서에 포함되어 있는 이미지 자체에 워터마크를 삽입함으로써 데이터의 변경을 막을 수 있었고, 그 변경을 검출할 수 있었으며, 텍스트 문서가 변경되었을 때 복원할 수 있는 기능을 가지게 하였다.

본 수법에 의하면 전자문서의 기술 내용이 부정적으로 변경되어도 증명서에 포함된 이미지로부터 텍스트 문서를 복원할 수 있다.

본 논문에서 연구하여 개발한 텍스트 임베딩 알고리즘은 분류하면 취약한 워터마크(Fragile Watermark)에 속한다.

현재 디지털 컨텐츠의 지작권을 보호하기 위해서 여러 가지 기법이 새로운 대안으로 떠오르

고 있지만 아직까지 모든 용도의 요구조건을 만족시키는 방법은 없다. 증명서의 인증은 아주 중요한 과정으로서 확고한 인증과 무결성을 보장해야 된다.

본 연구의 취약점은 텍스트 문서(증명서)의 저작권을 보호하기 위하여 그 문서에 이미지가 포함되어야 한다는 것이다. 실지 삽입한 텍스트 문서를 인증하기 위하여 필요한 다른 한 형태인 강인성을 필요로 할 경우도 있다. 기존의 텍스트 문서만을 이용하여 워터마크를 삽입할 수 있는 방법의 탐색이 필요하다.

참고문헌

[1] Munetoshi IWAKIRI, Yoshio MURAKAMI, " *Signature Seal with Recoverable Function of Text for Electronic Documents*" SCIS2000 The 2000 Symposium on Cryptography and Information Security Okinawa, Japan, January 26-28, 2000

[2] Brassil,J.T., Low,S.W., Maxemchuk,N.F. and O'Gorman,L.: *Electoronic marking and identification techniques to discourage document copying*, Proc.of IEEE INFOCOM'94, vol.3, pp.1278-1287(1994)

[3] Hecht,D.L.: *Embedded data glyph technology for hardcopy digital documents*, Proc.of SPIE, vol.2171, Color Hardcopy and Graphic Arts III, PP.341-352(1994)

[4] Tatsuro IKEDA, Kenichiro AKAI, Tsutomu MATSUMOTO, "*A Method of Text-Based Fingerprintion and Its Robustness*" SCISs2000 The 2000 Symposium on Cryptography and Information Security Okinawa, Japan, January 26-28, 2000