이미지 워터마킹과 스테가노그래피를 이용한 정보 은닉 시스템

팀 명 : 5조 Block's

지도교수 : 유 승 재 교수

조 장 : 김 태 윤

신 용 하

최 동 규

김 민 수

2016 . 05

중부대학교 정보보호학과

목 차

1. 서론

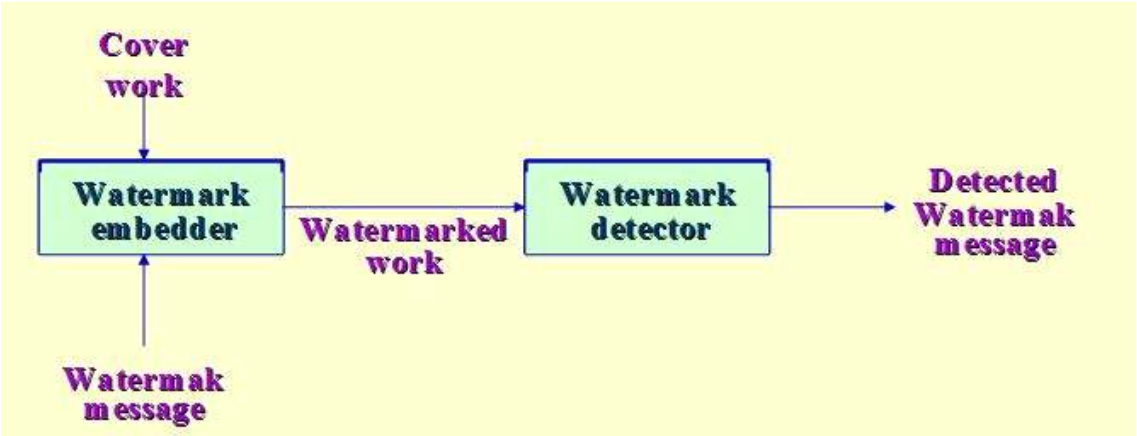
최근 인터넷 환경의 급속한 성장으로 인해 효율적인 디지털 콘텐츠 보급이 가능하게 되었다. 하지만 악의적인 공격자에 의한 저작권 침해 등으로 인해, 이미지 데이터 보호 및 비밀통신 방법에 관한 요구사항 또한 높아지고 있다. 따라서 디지털 콘텐츠의 저작권을 보호하고
비밀통신을 가능하게 하는 방법 중 하나인 '워터마킹'과 '스테가노그래피'기법에 대한 이론 확립과 프로그램의 구현 목표로 한다.

2. 관련 연구

1) 워터마킹

워터마킹은 위조지폐를 방지하기 위해 지폐 제작과정 중 젖어 있는 상태에서 그림(마크)을 넣는 기술이며 그래픽, 비디오, 오디오 등 모든 디지털 콘텐츠에 저작권 보호를 위해 이와 같은 특수처리를 하는 기술입니다. 이 때 삽입되는 저작권, 소유정보 또는 원본여부를 확인할 수 있도록 숨겨놓은 데이터, 사용권한을 부여 받은 사용자의 ID등의 식별 정보작권자임을
입증할 수 있는 정보가 삽입돼 있습니다. 이것은 나중에 저작권 분쟁이 발생하거나 시비가 있을 경우 콘텐츠에 있는 워터마크를 추출함으로써 저작권 문제를 해결할 수 있습니다.

- 개념도



Embedder 2입력

- 워터마크를 인코드 하려는 메시지
- 마크를 넣으려는 cover work

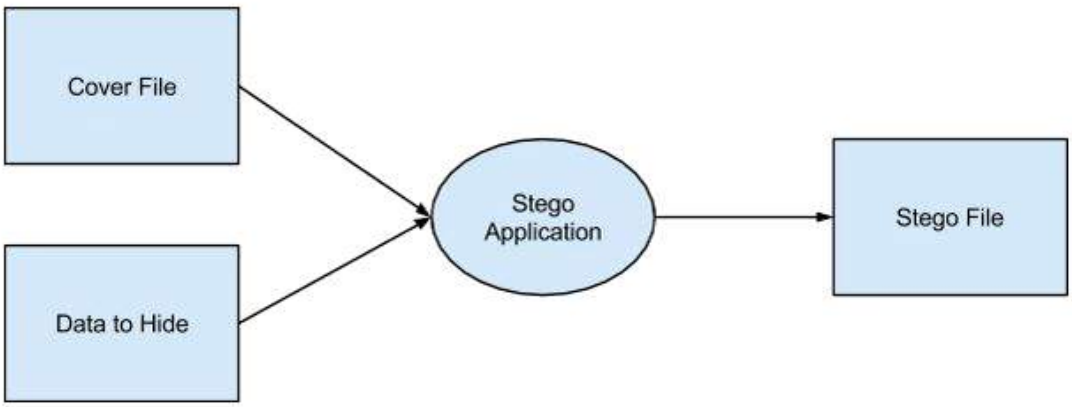
Watemark Emvedder의 출력을 저장 및 전송

Watemark detector는 워터마크가 있는지 확인하고 워터마크로 인코드 된 메시지를 출력

2) 스테가노그래피(Steganography)

디지털 콘텐츠의 저작권 보호와 정보보호의 방법으로, 멀티미디어 데이터의 제작자나, 저작권 관련 정보를 전달하려는 기밀 정보를 이미지 파일이나 MP3 파일 등에 암호화해 숨기는 심층암호 기술입니다. 비밀통신의 유형에는 일반적으로 전달하고자 하는 정보의 비밀(은닉성) 유지 방식에 따라 정보가 담긴 메시지가 드러나지 않도록 완전히 숨겨서 전달하는 '은폐' 방식과 메시지를 미리 정해진 약속에 따라 다른 형태로 '변형'해서 전달하는 방식이 있는데, 비밀 메시지 존재 자체를 '은폐'해서 전달하는 대표적 방식이 스테가노그래피 방식입니다.

이 기법은 일반적으로 커버이미지의 픽셀 (pixel) 값을 변경하여 다른 사람들이 비밀정보의 은닉 유무를 인지할 수 없도록 합니다.



3) AES 암호 알고리즘

DES의 안전성에 대한 여러 가지 공격 방법들이 발표되면서 미국의 NIST에서는 1998년 차세대 블록 암호 알고리즘인 AES를 공모하였습니다. 그 후 2년간의 심사과정을 거쳐 2000년 10월에 Rijindael 알고리즘이 AES 알고리즘으로 선정되었으며, 2001년 11월 FIPS-197로 등록되었습니다. 처음에 개발된 Rijindael 알고리즘은 암호화에 사용하는 키의 길이와 입력 평문의 길이다 128비트, 196비트, 256비트 중 하나를 선택 할 수 있었으나 FIPS-197에 등록된 AES 알고리즘은 입력 평문의 길이만 128비트, 사용하는 암호화 키의 길이만 128비트, 196비트,256비트 중에서 선택할 수 있도록 정의하고 있습니다. AES는 지금까지 알려진 블록 암호 알고리즘에 대한 모든 공격 방법들에 대해 안전하도록 설계되었으며, 하드웨어나 소프트웨어 구현 시 속도나 코드 Compactness면에서 효율적입니다.

- AES 암호 방식은 평문을 128비트 단위로 나누어 암호화, 복호화를 수행하며, 각각의 128 비트를 4x4 행렬로 표현하여 연산을 수행합니다. 아래 그림과 같이 4x4행렬로 표현 된 암호화, 복호화 과정의 중간 결과를 state라 하며, 행렬의 각 열의 32비트를 워드라고 합니다.

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

AES의 기본 연산은 바이트 단위로 수행되며, 각 바이트를 유한체 GF(2^8) 위의 다항식으로 표현하여 연산을 수행하게 됩니다. 암호화, 복호화 과정은 바이트 단위의 덧셈, 곱셈 연산으로 이루어져 있습니다. AES의 암호화, 복호화 과정에서의 바이트들의 덧셈 연산은 비트 단위 XOR을 의미하며 바이트들의 곱셈 연산은 mod 8차 기약 다항식에서의 곱셈을 의미합니다.

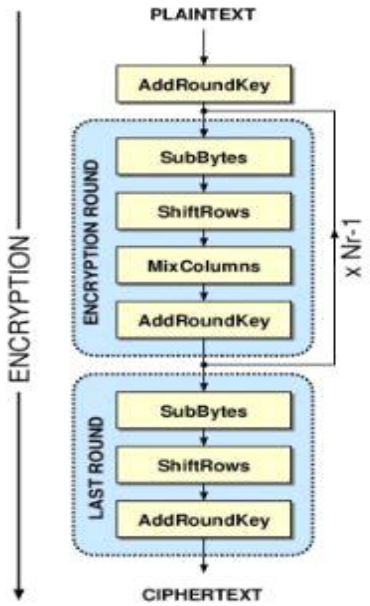
- 암호화 과정

AES의 암호화 과정의 각 라운드는 비 선형성(함수의 값이 독립변수의 값과 비례관계에 있지 않는 것)을 가지는 S-Box를 적용하여 바이트 단위로 치환을 수행하는 SubBytes()연산, 행 단위로 순환 시프트를 수행하는 ShiftRow()연산, 높은 확산(Diffusion)을 제공하기 위해 열 단위로 혼합(mixing)하는 MixColumns()연산과 마지막으로 라운드 키와 state를 XOR하는 AddRoundKey()연산으로 구성됩니다.

사용하는 암호화 키의 길이에 따라, 암,복호화 과정에 필요한 라운드 수는 다음 표와 같이 정의 합니다.

	키 길이 (Nk Words)	블록 길이 (Nb words)	라운드 수 (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

AES의 암호화 과정은 DES와는 달리, 첫 번째 라운드를 수행하기 전에 먼저 초기 평문과 라운드 키의 XOR 연산을 수행하므로, 암호화 과정에 필요한 전체 라운드 키의 개수는 Nr+1 개가 됩니다. 그리고, 암호화의 마지막 라운드에서는 MixColumn() 연산을 수행하지 않는다는 특징이 있습니다.



SubBytes()

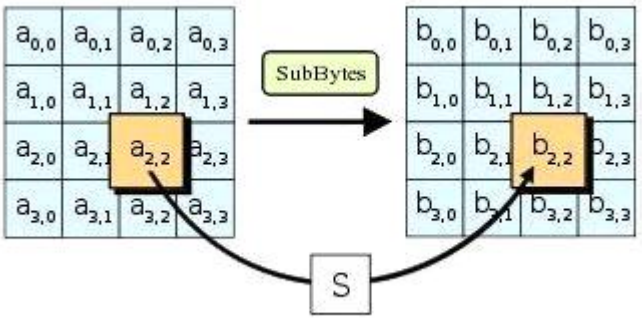
- SubBytes() 연산 과정은 암호문이 비 선형성을 갖도록 하기 위해 바이트 단위로 역 변환이 가능한 S-Box를 적용하는 것입니다. 연산은 크게 두 단계로 구성됩니다.

- 첫 번째 단계는, 각 바이트를 $GF(2^8)$ 위의 다항식으로 표현하여 mod 8차 기약다항식 상에서의 역수를 구하는 것으로 확장 유클리드 호제법을 이용하여 수행하게 됩니다.

- 다음은 GF(2) 위에서 Affine변환은 적용하는 과정으로 다음과 같은 행렬을 이용합니다.

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 10001111 \\ 11000111 \\ 11100011 \\ 11110001 \\ 11111000 \\ 01111100 \\ 00111110 \\ 00011111 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

두 단계를 통해 SubBytes() 연산을 수행하면 현재 state의 각 바이트는 아래 그림과 같이 다른 바이트로 치환 됩니다.

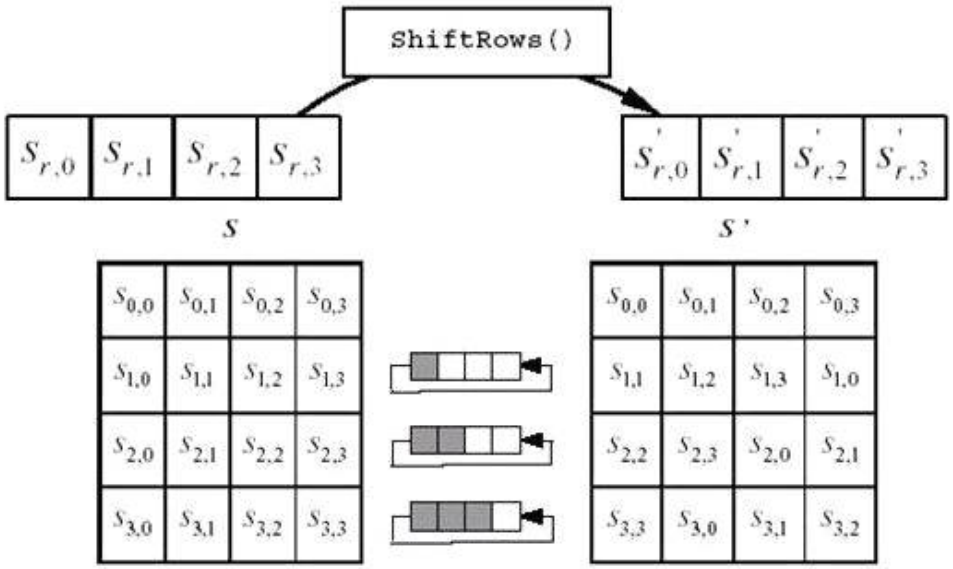


Affine 변환의 결과 집합을 S-Box로 표현하면 다음과 같습니다.

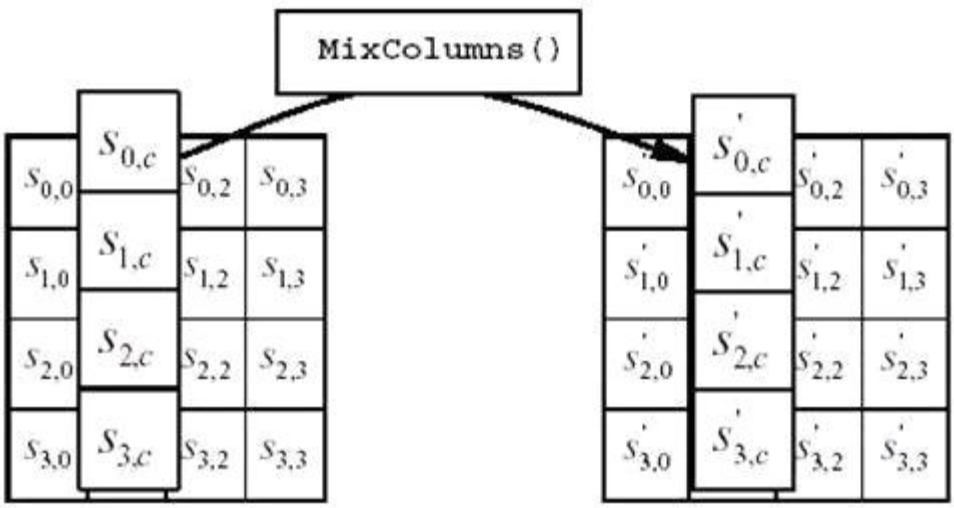
		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

ShiftRows()

- ShiftRows() 연산은 state의 각 행 단위로 정해진 수만큼 순환 시프트를 수행하는 것입니다. 이 때, state의 0번째 행은 그대로 두고, 1번째 행은 1번, 2번째 행은 2번, 3번째 행은 3번 왼쪽으로 순환 시프트를 수행합니다.



MixColumns()



MixColumns() 연산은 state의 각 열을 4개의 항을 갖는 3차 다항식으로 표현하고, mod x^4+1 상에서 다항식 a(x) = 03 · x^3 + x^2 + x + 02와 곱하는 과정입니다.

- 다항식 a(x)와 b(x)를 mod x^4+1 상에서 곱한 결과는 다음과 같은 행렬의 곱셈으로 표현할 수 있습니다.

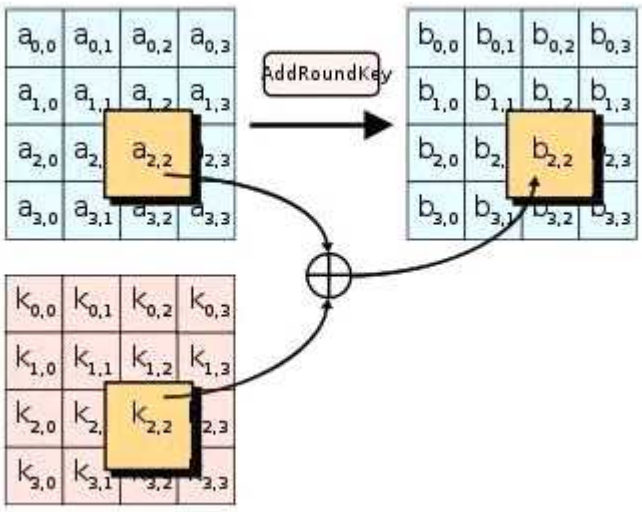
$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

AddRoundKey()

- AddRoundKey() 연산은 라운드 키와 현재 state를 비트 단위로 XOR를 수행하는 과정입니다. 암호화 과정의 state와 라운드 키는 동일한 크기를 가지며, 1 라운드를 수행하기 전에 초기 평문과 라운드 키를 XOR하는 과정이 필요하므로, AddRoundKey() 연산은 전체 암호화 과정에서 Nr+1번 수행하게 됩니다.

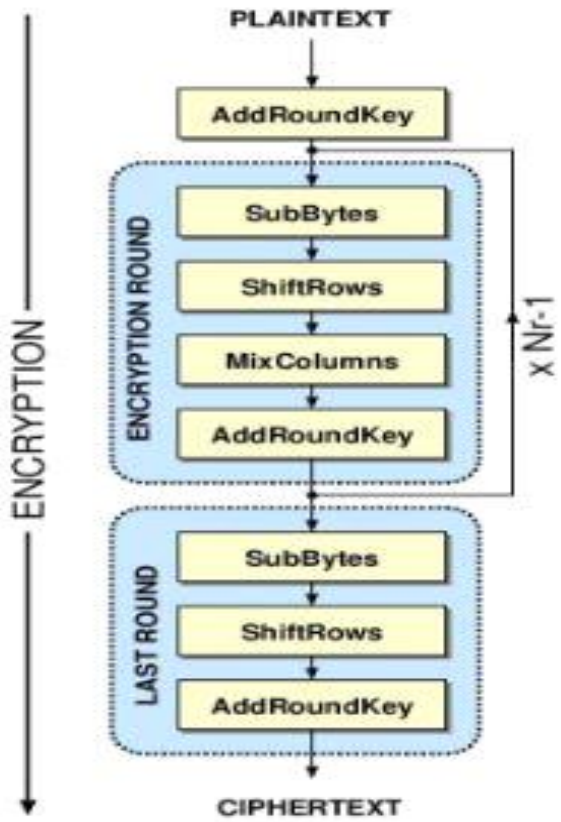
- 각 라운드에서 사용하는 라운드 키는 암호화 키로부터 생성되며, 자세한 라운드 키 생성 과정은 키 확장에서 설명합니다.

- 현재 state와 라운드 키의 AddRoundKey() 연산과정은 다음 그림과 같습니다.



- 복호화 과정

AES의 복호화 과정은 암호화 과정의 역 변환으로 InvSubBytes(), InvShiftRows(), InvMixColumns(), AddRoundKey() 연산으로 구성됩니다. 암호화 과정에서 마지막 라운드는 이전의 라운드들과 달리 MixColumns() 연산을 포함하지 않으므로, 복호화 과정의 첫번째 라운드가 이후의 라운드들과 달리 InvMixColumns() 과정을 포함하지 않습니다.



[암호화 과정에서 역방향으로 다시 올라간다고 생각하면 됩니다]

복호화 과정의 첫번째 라운드를 제외한 각 라운드는 AddRoundKey(), InvMixColumns(), InvShiftRows(), InvSubBytes() 순서로 연산을 수행하며, 라운드 키는 암호화의 역순으로 Nr 번째 라운드 키부터 사용합니다.

InvSubBytes()

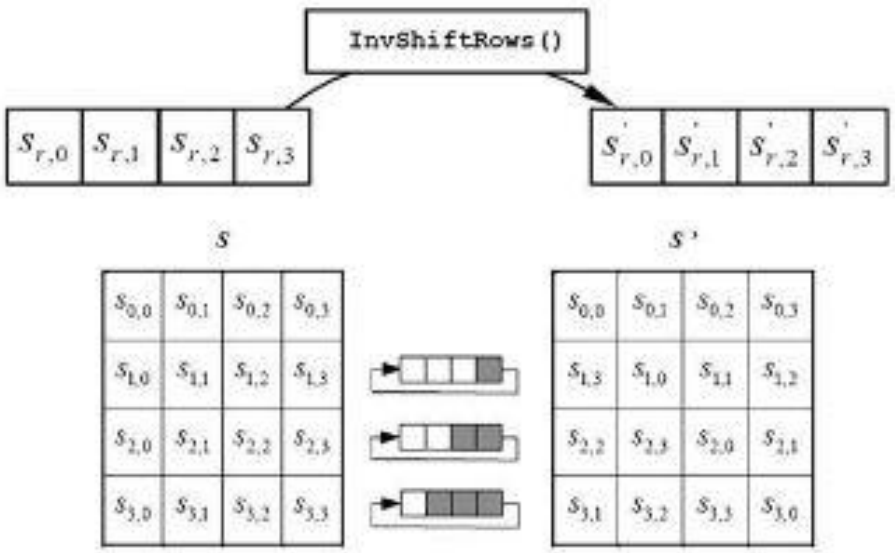
- InvSubBytes() 연산은 S-Box를 이용한 바이트 단위의 치환 연산인 SubBytes()의 역 변환이다. 따라서, SubBytes()에서 적용한 Affine 변환의 역변환을 적용한 후, 각 바이트 단위로 mod 8차 기약 다항식 상에서의 역수를 구하는 과정으로 이루어지게 됩니다.

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

[AES의 Inverse S-Box]

InvShiftRows()

- InvShiftRows() 연산은 ShiftRows()의 역 변환으로, ShiftRows() 과정에서 수행한 순환 시프트 회수 만큼 다시 오른쪽으로 시프트 연산을 수행하는 과정입니다.



InvMixColumns()

- InvMixColumns() 연산은 MixColumns() 연산의 역 변환으로 state의 열 단위로 3차 다항식을 곱하는 과정입니다. 즉, MixColumns() 연산에서 사용한 다항식의 역 다항식을 mod x^4+1 상에서 곱하는 것입니다.

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

AddRoundKey()

- 복호화 과정의 AddRoundKey() 연산은 암호화 과정과 마찬가지로, 현재 state와 라운드 키를 비트 단위로 XOR을 수행하는 것입니다.

3. 개발 환경

1) C#

모든 것을 객체로 취급하는 컴포넌트 프로그래밍언어로, 시샵(C-sharp)이라고 발음합다. 2000년 6월 마이크로소프트가 닷넷(.NET) 폴랫폼을 위해 개발하였습니다. C++(시플러스플러스)에 기본을 둔 언어로, 비주얼베이직이나 자바(Java)와도 비슷하다. 따라서 비주얼베이직과 자바·C++ 등의 장점을 지닌다. 곧 비주얼 언어가 가진 사용자 친화성, C++의 객체지향성, 자바의 분산환경처리에 적합한 다중성 등을 모두 지니는 컴포넌트 기반의 소프트웨어 개발 패러다임을 반영합니다.
웹을 통해 정보와 서비스를 교환하고, 개발자들이 이식성(portability) 높은 응용프로그램들을 만들어 낼 수 있게 고안되었습니다. 즉, 이 프로그래밍언어를 사용하면 대대적인 개정 없이 하나 이상의 OS(운영체제)에서 사용될 수 있는 응용프로그램들을 만들어낼 수가 있습니다.
따라서 프로그래머가 별도의 코드를 만들지 않고서도 새로운 제품이나 서비스를 빠르고 값싸게 시장에 내놓을 수 있게 됩니다.

2) MS-sql

MS-SQL이란 일종의 운영자가 웹브라우저를 관리함에 있어 사용자의 회원관리와 쇼핑몰에서 물품목록을 리스트별로 볼 수 있도록 구현을 해주고 DATABASE관리를 해주는 Microsoft사에서 만든 웹DB라고 할 수 있습니다.
웹상에서 회원들을 관리 할 때 회원들의 정보를 DB에 저장했다가 DB에서 회원들의 정보를 다시 불러와서 리스트 형식으로 출력해주는 역할을 합니다.
그리고 쇼핑몰에서 사용자가 물건을 주문 한 경우와 물건을 반품시킨 경우, 이러한 일련의 과정들을 웹상에서 확인 할 수 있도록 구현해 주는 것이 웹DB라고 할 수 있습니다.
웹DB의 종류는 여러가지가 있지만 윈도우 형식으로 관리자가 편리하게 만들고 구현을 할수 있는게 MS-SQL만의 특징이라 할 수 있고, 또 MS-SQL은 다른 웹DB들과는 달리 엔터프라이즈 관리자라는 관리도구가 갖추어져 있어 DB를 쉽게 만들고 삭제 할 수 있는 기능이 있습니다. PHP에서도 PHPMYADMIN이라는 관리도구가 있기는 한데 MS-SQL의 EM만큼은 못하지만 EM의 큰 특징은 윈도우상에서 클릭 몇 번만으로 DB를 손쉽게 만들 수 있다는 장점이 있습니다. 반면에 단점은 비용이 많이 들어가 있고, 엔터프라이즈 관리자 하나를 돌리는데 Microsoft사에다 상당의 비용을 주고 이용을 해야 하며 호스팅 비용에서도 몇 배의 차이를 보입니다.

3) MS-access

마이크로소프트 액세스(Microsoft Access)는 마이크로소프트 오피스에 포함된 데이터베이스 프로그램이다. 1992년 11월 13일 액세스 1.0 버전으로 처음 개발되었으며 현재 최신 버전은 마이크로소프트 오피스 액세스 2016이다. 2003 버전까지는 .mdb 확장자를 기본으로 사용했으나 2007 버전부터는 .accdb를 사용하고 있습니다. 액세스는 대기업의 부서와 프로그래머를 포함하여 소형 비즈니스에서 사용됩니다. 프로그래머의 입장에서 액세스에서 볼 수 있는 이점들 가운데 하나는 SQL과 연동이 잘된다는 점입니다. 또한 액세스는 마이크로소프트 인터넷 정보 서비스(IIS)와 액티브 서버 페이지(ASP) 위에서 기본 웹 기반의 응용 프로그램들을 위한 데이터베이스로 사용될 수 있습니다.

4. 프로그램 구현 및 소스

1. Stegano / Watermark / Datdbase 선택 창
2. 저작권 정보 입력
3. 암호화 복호화

Stegano 구현

Cover Image 선택 후, 저작권 정보 or Hide Image를 등록
Encrypt로 암호화 , Decrypt로 복호화
DB 버튼 클릭 – DB 저장

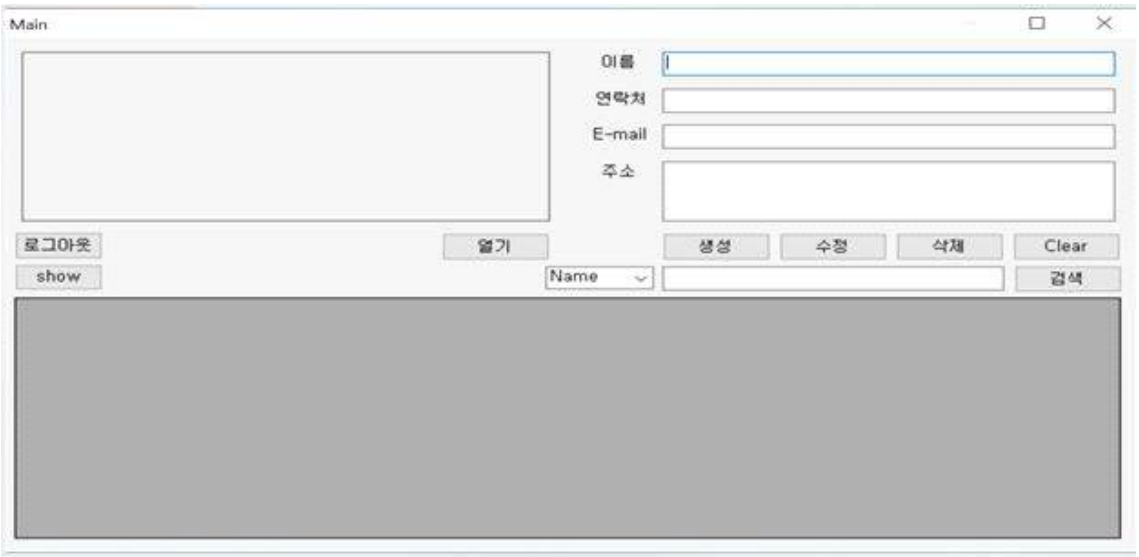
Watermark 구현 [기본 창]

텍스트 입력 후 위치,투명도 선택

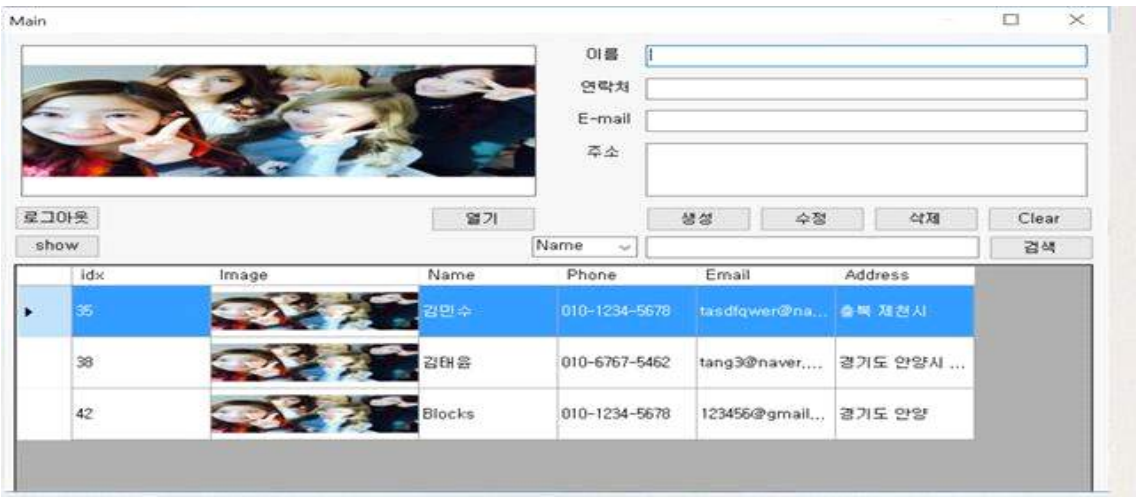
미리보기 화면

워터마크 입력 후 저장 한 화면

Datdbase 구현



처음에 저장 된 이미지와 개인정보를 불러 올 수 있으며
새롭게 생성, 수정 할 수 있다



기본 시스템 소스

```
using System;
using System.Collections.Generic;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Drawing.Imaging;
using System.Drawing.Drawing2D;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO;
using System.Security.Cryptography;
using System.Data.SqlClient;
using System.Threading;
namespace Block
{
public partial class Form1 : Form
    {
public String key = "abcdefghijklmnopqrstuvwxyz123456"

AES256Cipher aes;

public Form1()
{
InitializeComponent();
myWatermarkColor = Color.SteelBlue;
optTop.Checked = true
myFont = txtWaterMark.Font;

aes = new AES256Cipher();
}

        #region stegano

private byte getByte(byte[] bits)
{
String bitString = ""

for (int i = 0; i < 8; i++)
```

```
bitString += bits[i];

byte newpix = Convert.ToByte(bitString, 2);
int dePix = (int)newpix;
return (byte)dePix;
}

private byte[] getBits(byte simplepixel)
{
int pixel = 0;
pixel = (int)simplepixel;
BitArray bits = new BitArray(new byte[] { (byte)pixel });
bool[] boolarray = new bool[bits.Count];
bits.CopyTo(boolarray, 0);
byte[] bitsArray = boolarray.Select(bit => (byte)(bit ? 1 : 0)).ToArray();
Array.Reverse(bitsArray);
return bitsArray;
}

private void button1_Click(object sender, EventArgs e)
{
OpenFileDialog imgopen = new OpenFileDialog();
imgopen.Filter = "Bmp files(.bmp)|*.bmp"
imgopen.InitialDirectory = "C:₩₩Users₩₩tang₩₩Pictures₩₩Saved Pictures"
if (imgopen.ShowDialog() == DialogResult.OK)
{
imgpath.Text = imgopen.FileName.ToString();
img.ImageLocation = imgpath.Text;
name1.Clear();
phone.Clear();
email.Clear();
addr.Clear();
}

}

private void restore_bits(ref string result, byte[] input_byte)
{
for (int i = 0; i < input_byte.Length; i++)
result += input_byte[i];
}

private byte[] chck_encryption_key(byte[] input, int length)
```

```
{
byte[] result = new byte[length];
for (int i = 0; i < length; i++)
{
if ((int)input[i] < 64) result[i] = 0;
else result[i] = 1;
}
return result;
}

private Bitmap injection_byte(byte[] input, ref Bitmap image, int pos_x, int pos_y)
{
byte[] AlphaBits, RedBits, GreenBits, BlueBits;
byte newAlpha, newRed, newGreen, newBlue;
Color pixel;

string test = ""
pixel = image.GetPixel(pos_x, pos_y);

AlphaBits = getBits((byte)pixel.A);
RedBits = getBits((byte)pixel.R);
GreenBits = getBits((byte)pixel.G);
BlueBits = getBits((byte)pixel.B);

RedBits[5] = input[0];
RedBits[6] = input[1];
RedBits[7] = input[2];
GreenBits[6] = input[3];
GreenBits[7] = input[4];
BlueBits[5] = input[5];
BlueBits[6] = input[6];
BlueBits[7] = input[7];

newAlpha = getByte(AlphaBits);
newRed = getByte(RedBits);
newGreen = getByte(GreenBits);
newBlue = getByte(BlueBits);

test = "test = "
for (int i = 0; i < input.Length; i++) test += input[i];

image.SetPixel(pos_x, pos_y, Color.FromArgb((int)newAlpha, (int)newRed, (int)newGreen,
(int)newBlue));
```

```
AlphaBits = read_byte(ref image, pos_x, pos_y);
test = " test2 = "
for (int i = 0; i < AlphaBits.Length; i++) test += AlphaBits[i];
return image;
}

private int Inject_chk_byte(byte[] chk_byte, ref Bitmap image, int max_posx, int
max_posy)
{
const byte save_size = 8;
int size = (chk_byte.Length / save_size);

if (chk_byte.Length % save_size > 0) size += 1;

string test = "test = ", test2 = "test2 = "
int i, j;

byte[] new_chk_bytes = new byte[save_size];
byte[] read_bytes = new byte[save_size];

for (i = 0; i < size; i++)
{
test = "test1 = "
test2 = "test2 = "
for (j = 0; j < save_size; j++)
{
if (j + i * save_size >= chk_byte.Length)
break

new_chk_bytes[j] = chk_byte[j + i * save_size];
test2 += new_chk_bytes[j];
}
image = injection_byte(new_chk_bytes, ref image, (byte)(max_posx - 1),
(byte)(max_posy - (i + 1)));
read_bytes = read_byte(ref image, (byte)(max_posx - 1), (byte)(max_posy - (i + 1)));
for (j = 0; j < read_bytes.Length; j++)
{
test += read_bytes[j];
}
}
return size;
}
```

```
private byte[] read_byte(ref Bitmap image, int pos_x, int pos_y)
{
const byte save_size = 8;
byte[] result = new byte[save_size];

byte[] AlphaBits, RedBits, GreenBits, BlueBits;
Color pixel;

pixel = image.GetPixel(pos_x, pos_y);

AlphaBits = getBits((byte)pixel.A);
RedBits = getBits((byte)pixel.R);
GreenBits = getBits((byte)pixel.G);
BlueBits = getBits((byte)pixel.B);

result[0] = RedBits[5];
result[1] = RedBits[6];
result[2] = RedBits[7];
result[3] = GreenBits[6];
result[4] = GreenBits[7];
result[5] = BlueBits[5];
result[6] = BlueBits[6];
result[7] = BlueBits[7];

return result;
}

private string save_file(Bitmap image)
{
SaveFileDialog savefile = new SaveFileDialog();
savefile.Filter = "Bmp files(.bmp)|*.bmp"
savefile.InitialDirectory = "C:₩₩Users₩₩tang₩₩Pictures₩₩Saved Pictures"
if (savefile.ShowDialog() == DialogResult.OK)
{
imgpath.Text = savefile.FileName.ToString();
img.ImageLocation = imgpath.Text;
image.Save(imgpath.Text);
return imgpath.Text;
}

return ""
}
```

```
private string restore_byte_string(string byte_string)
{
string b_string, result = ""
int i = 0, j = 0;
for (i = 0; i * 8 + j < byte_string.Length; i++)
{
b_string = ""
for (j = 0; j < 8 && i * 8 + j < byte_string.Length; j++)
{
b_string += byte_string.Substring(i * 8 + j, 1);

}

if (b_string.Length < 8)
{
string zero = ""
for (j = 0; (8 - b_string.Length) - j > 0; j++)
{
zero += "0"
}
b_string = zero + b_string;
}
result += (char)Convert.ToByte(b_string, 2);
}
return result;
}

private void init_text_box(string d_str)
{
string[] str = d_str.Split('#');
name1.Text = str[0];
phone.Text = str[1];
email.Text = str[2];
addr.Text = str[3];

}

private void Encryption_Click_1(object sender, EventArgs e)
{
if (imgpath.Text == "" || name1.Text == "" || phone.Text == "" || email.Text == "" ||
addr.Text == "")
MessageBox.Show("정보를입력하세요);
else
```

```
                {
richTextBox1.Clear();
string encode = aes.AES_encrypt(name1.Text + "#" + phone.Text + "#" + email.Text
+ "#" + addr.Text, key);
richTextBox1.Text = (encode);

byte[] chk_bytes = chck_encryption_key(Encoding.UTF8.GetBytes(encode),
encode.Length);

string chk_string = ""
for (int k = 0; k < encode.Length; k++) chk_string += chk_bytes[k];

int i;

byte[] MsgBits;
byte[] r_byte = new byte[8];
Bitmap simple = new Bitmap(imgpath.Text);
i = 0;
Color pixel;
while (i < richTextBox1.TextLength)
{
pixel = simple.GetPixel(i, i);
char letter = Convert.ToChar(richTextBox1.Text.Substring(i, 1));
byte value = Convert.ToByte(letter);
MsgBits = getBits((byte)value);
//for (int k = 0; k < MsgBits.Length; k++) input_bits += MsgBits[k];
                        injection_byte(MsgBits, ref simple, (byte)i, (byte)i);
i++;
}
pixel = simple.GetPixel(simple.Width - 1, simple.Height - 1);
simple.SetPixel(simple.Width - 1, simple.Height - 1, Color.FromArgb(pixel.R, pixel.G,
richTextBox1.TextLength));


SaveFileDialog savefile = new SaveFileDialog();
savefile.Filter = "Bmp files(.bmp)|*.bmp"
savefile.InitialDirectory = "C:₩₩Users₩₩tang₩₩Pictures₩₩Saved Pictures"
if (savefile.ShowDialog() == DialogResult.OK)
{
imgpath.Text = savefile.FileName.ToString();
img.ImageLocation = imgpath.Text;
simple.Save(imgpath.Text);
}
```

```
richTextBox1.Clear();
}
}

private void Decryption_Click_1(object sender, EventArgs e)
{
int i;
Bitmap img = new Bitmap(imgpath.Text);
Color lastpixel = img.GetPixel(img.Width - 1, img.Height - 1);
int msgLength = lastpixel.B;
byte[] BitsToDecrypt = new byte[8];
string message = ""

byte[] read_bytes;
i = 0;
Color pixel2;
while( i<msgLength && i < img.Width && i< img.Height )
{
pixel2 = img.GetPixel(i, i);
read_bytes = read_byte(ref img, i, i);
byte value = getByte(read_bytes);
message += (char)value;
i++;
}

if (Convert.ToInt32(message) > 127)
{
MessageBox.Show("암호화된이미지가아닙니다);
}
else
                {
richTextBox2.Text = message;
MessageBox.Show(message);

richTextBox2.Text = aes.AES_decrypt(richTextBox2.Text, key);
init_text_box(richTextBox2.Text);
richTextBox2.Clear();
}
}

private void clear1_Click_1(object sender, EventArgs e)
{
richTextBox1.Clear();
```

```
}

private void clear2_Click_1(object sender, EventArgs e)
{
name1.Clear();
phone.Clear();
email.Clear();
addr.Clear();
}

        #endregion

        #region watermark

        #endregion

#region login

private void ID_TextChanged(object sender, EventArgs e)
{

}


private void button4_Click(object sender, EventArgs e)
{
SqlConnection con = new SqlConnection(@"Data Source=.₩SQLEXPRESS;AttachDbFilename=C:₩Users₩tang₩Documents₩Data.mdf;Integrated Security=True;Connect Timeout=30;User Instance=True");
SqlDataAdapter sda = new SqlDataAdapter("Select Count(*) From Login where Username= '" + ID.Text + "' and Password ='" + PASS.Text + "'", con);
DataTable dt = new DataTable();
sda.Fill(dt);
if (dt.Rows[0][0].ToString() == "1")
{
this.Hide();
Main ss = new Main();
ss.Show();

}
else
            {
MessageBox.Show("아이디와비밀번호를다시입력하세요);
```

```
}
}

private void button5_Click(object sender, EventArgs e)
{
SqlConnection con = new SqlConnection(@"Data Source=탱?3;Initial Catalog=Employee;Integrated Security=True");
SqlCommand command;
string imgLoc = imgpath.Text;
if (!(name1.Text == "" || phone.Text == "" || email.Text == "" || addr.Text == ""))
{
try
                    {
byte[] img = null
FileStream fs = new FileStream(imgLoc, FileMode.Open, FileAccess.Read);
BinaryReader br = new BinaryReader(fs);

img = br.ReadBytes((int)fs.Length);
string sql = "INSERT INTO Employee (Image, Name, Phone, Email, Address) VALUES (@img,'" + name1.Text + "','" + phone.Text + "','" + email.Text + "','" + addr.Text + "')"

if (con.State != ConnectionState.Open)
con.Open();

command = new SqlCommand(@sql, con);

command.Parameters.Add(new SqlParameter("@img", img));
int x = command.ExecuteNonQuery();

MessageBox.Show("DB에저장되었습니다);
name1.Text = ""
phone.Text = ""
email.Text = ""
addr.Text = ""
}
catch (Exception ex)
{
con.Close();
MessageBox.Show(ex.Message);
}
}
else
```

```
                    MessageBox.Show("정보를다시입력하세요);
}

private void Form1_Load(object sender, EventArgs e)
{
// TODO: 이 코드는 데이터를 테이블에 로드합니다 필요한 경우 이 코드를 이동하거나
제거할수있습니다

          }

private void tabPage1_Click(object sender, EventArgs e)
{

}

private void richTextBox1_TextChanged(object sender, EventArgs e)
{

}

private void email_TextChanged(object sender, EventArgs e)
{

}

private void name1_TextChanged(object sender, EventArgs e)
{

}

private void tabPage2_Click(object sender, EventArgs e)
{

}
          #endregion

#region Member Variables

System.Drawing.Color myWatermarkColor;
System.Drawing.Font myFont;
          #endregion

          #region Constructor
```

```
private void btnSave_Click(object sender, EventArgs e)
{
SaveFileDialog save = new SaveFileDialog();
Bitmap ImageFile = null
save.Filter = "Bmp files(.bmp)|*.bmp"
save.InitialDirectory = "C:₩₩Users₩₩tang₩₩Pictures₩₩Saved Pictures"
if (this.picContainer.Image != null)
{
if (save.ShowDialog() == DialogResult.OK)
{
ImageFile = (Bitmap)this.picContainer.Image;
ImageFile.Save(save.FileName, ImageFormat.Bmp);
}
}
}
private void btnPreview_Click(object sender, EventArgs e)
{
picContainer.Image = Image.FromFile(imgpath2.Text);
int opac = 0;
string sOpacity = cboOpacity.Text;

switch (sOpacity)
{
case "100%":
opac = 255;
break
case "75%":
opac = 191;
break
case "50%":
opac = 127;
break
case "25%":
opac = 64;
break
case "10%":
opac = 25;
break
case "0%":
opac=0;
break
default:
opac = 127;
```

```
break
}
Graphics g = Graphics.FromImage(picContainer.Image);
Brush myBrush = new SolidBrush(Color.FromArgb(opac, myWatermarkColor));

SizeF sz = g.MeasureString(txtWaterMark.Text, myFont);
int X;
int Y;
if (optLT.Checked == true)
{
X = 0;
Y = 0;
g.DrawString(txtWaterMark.Text, myFont, myBrush, new Point(X, Y));
}
else if (optRT.Checked == true)
{
X = (int)(picContainer.Image.Width - sz.Width);
Y = 0;
g.DrawString(txtWaterMark.Text, myFont, myBrush, new Point(X, Y));
}
else if (optLB.Checked == true)
{
X = 0;
Y = (int)(picContainer.Image.Height - sz.Height);
g.DrawString(txtWaterMark.Text, myFont, myBrush, new Point(X, Y));
}
else if (optRB.Checked == true)
{
X = (int)(picContainer.Image.Width - sz.Width) ;
Y = (int)(picContainer.Image.Height - sz.Height);
g.DrawString(txtWaterMark.Text, myFont, myBrush, new Point(X, Y));
}
else if (optTop.Checked == true)
{
X = (int)(picContainer.Image.Width - sz.Width) / 2;
Y = 0;
g.DrawString(txtWaterMark.Text, myFont, myBrush, new Point(X, Y));
}
else if(optBottom.Checked == true)
{
X = (int)(picContainer.Image.Width - sz.Width) / 2;
Y = (int)(picContainer.Image.Height - sz.Height);
g.DrawString(txtWaterMark.Text, myFont, myBrush, new Point(X, Y));
```

```
}
else if (optCenter.Checked == true)
{
X = (int)(picContainer.Image.Width - sz.Width) / 2;
Y = (int)(picContainer.Image.Height - sz.Height) / 2;
g.DrawString(txtWaterMark.Text, myFont, myBrush, new Point(X, Y));
}
else if (optRight.Checked == true)
{
X = (int)(picContainer.Image.Width - sz.Width) ;
Y = (int)(picContainer.Image.Height - sz.Height) /2;
g.DrawString(txtWaterMark.Text, myFont, myBrush, new Point(X, Y));
}
else if (optLeft.Checked == true)
{
X = 0;
Y = (int)(picContainer.Image.Height-sz.Height)/2;
g.DrawString(txtWaterMark.Text, myFont, myBrush, new Point(X, Y));
}
else
                {
MessageBox.Show("위치를선택해주세요);
}
}
          #endregion

#region Dispose of the Application
private void exitToolStripMenuItem_Click(object sender, EventArgs e)
{
this.Dispose();
}
          #endregion
          private void fileToolStripMenuItem_Click(object sender, EventArgs e)
{
}
private void btnOpen_Click(object sender, EventArgs e)
{
OpenFileDialog imgopen2 = new OpenFileDialog();
imgopen2.Filter = "Bmp files(.bmp)|*.bmp"
imgopen2.InitialDirectory = "C:₩₩Users₩₩tang₩₩Pictures₩₩Saved Pictures"
if (imgopen2.ShowDialog() == DialogResult.OK)
{
imgpath2.Text = imgopen2.FileName.ToString();
```

```
picContainer.ImageLocation = imgpath2.Text;
}
}

private void picContainer_Click(object sender, EventArgs e)
{
}

private void optTop_CheckedChanged(object sender, EventArgs e)
{
}

private void panel2_Paint(object sender, PaintEventArgs e)
{
}

private void btnFont_Click(object sender, EventArgs e)
{
fontDialog1.ShowColor = true
fontDialog1.Font = txtWaterMark.Font;
fontDialog1.Color = txtWaterMark.ForeColor;

if (fontDialog1.ShowDialog() != DialogResult.Cancel)
{
myFont = fontDialog1.Font;
myWatermarkColor = fontDialog1.Color;
txtWaterMark.Font = fontDialog1.Font;
txtWaterMark.ForeColor = fontDialog1.Color;
}
}

private void Capa_Click(object sender, EventArgs e)
{
}
}
}
```

데이터 베이스

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
```

```
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Data.SqlClient;

using System.IO;

namespace Block
{
public partial class Main : Form
    {
SqlConnection con = new SqlConnection(@"Data Source=탱?3;Initial Catalog=Employee;Integrated Security=True");

SqlDataAdapter da = new SqlDataAdapter();

SqlCommand command;
string imgLoc = ""
public Main()

{
InitializeComponent();
}

private void textsearch_KeyPress(object sender, KeyPressEventArgs e)
{

}

private void dataGridView1_KeyDown(object sender, KeyEventArgs e)
{
if (e.KeyCode == Keys.Delete)
{
if (MessageBox.Show("Are you sure want to delete this record ?", "Message", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
employeeBindingSource.RemoveCurrent();
}
}

private void Brouse_Click(object sender, EventArgs e)
{
```

```
try
                {
OpenFileDialog dlg = new OpenFileDialog();
dlg.Filter = "Bmp Files (*.bmp)|*.bmp"
dlg.Title = "Select Employee Picture"
if (dlg.ShowDialog() == DialogResult.OK)
{
imgLoc = dlg.FileName.ToString();
pictureBox1.ImageLocation = imgLoc;
}
}
catch (Exception ex)
{
MessageBox.Show(ex.Message, "Message", MessageBoxButtons.OK,
MessageBoxIcon.Error);
employeeBindingSource.ResetBindings(false);
}
}
void showdata()
{
string sql = "SELECT idx, Image, Name, Phone, Email, Address FROM Employee
ORDER BY idx"
con.Open();
dataGridView1.RowTemplate.Height=50;
dataGridView1.Height = 250;
DataTable dt = new DataTable();

command = new SqlCommand(sql, con);
da.SelectCommand = command;
dt.Clear();
da.Fill(dt);

dataGridView1.DataSource = dt;
dataGridView1.Columns[1].Width = 150;
for(int i=0; i<dataGridView1.Columns.Count;i++)
{
if(dataGridView1.Columns[i] is DataGridViewImageColumn)
{
((DataGridViewImageColumn)dataGridView1.Columns[i]).ImageLayout =
DataGridViewImageCellLayout.Stretch;
}
}
con.Close();
```

```
clear();
}
void clear()
{
textname.Clear();
textphone.Clear();
textemail.Clear();
textaddress.Clear();
textname.Focus();
}

private void show_Click(object sender, EventArgs e)
{
string sql = "SELECT idx, Image, Name, Phone, Email, Address FROM Employee
ORDER BY idx"
con.Open();
dataGridView1.RowTemplate.Height=50;
dataGridView1.Height = 250;
DataTable dt = new DataTable();

command = new SqlCommand(sql, con);
da.SelectCommand = command;
dt.Clear();
da.Fill(dt);

dataGridView1.DataSource = dt;
dataGridView1.Columns[1].Width = 150;
for(int i=0; i<dataGridView1.Columns.Count;i++)
{
if(dataGridView1.Columns[i] is DataGridViewImageColumn)
{
((DataGridViewImageColumn)dataGridView1.Columns[i]).ImageLayout =
DataGridViewImageCellLayout.Stretch;
}
}

con.Close();
}

private void add_Click(object sender, EventArgs e)
{
try
        {
```

```
byte[] img = null
FileStream fs = new FileStream(imgLoc,FileMode.Open, FileAccess.Read);
BinaryReader br = new BinaryReader(fs);

img = br.ReadBytes((int)fs.Length);
string sql ="INSERT INTO Employee (Image, Name, Phone, Email, Address) VALUES (@img,'" + textname.Text + "','" + textphone.Text + "','" + textemail.Text + "','" + textaddress.Text + "')"

if(con.State != ConnectionState.Open)
con.Open();

command= new SqlCommand(@sql,con);

command.Parameters.Add(new SqlParameter("@img",img));
int x = command.ExecuteNonQuery();

MessageBox.Show("DB에저장되었습니다);
textname.Text = ""
textphone.Text = ""
textemail.Text = ""
textaddress.Text = ""

string sql2 = "SELECT idx, Image, Name, Phone, Email, Address FROM Employee ORDER BY idx"

dataGridView1.RowTemplate.Height = 50;
dataGridView1.Height = 250;
DataTable dt = new DataTable();

command = new SqlCommand(sql2, con);
da.SelectCommand = command;
dt.Clear();
da.Fill(dt);

dataGridView1.DataSource = dt;
dataGridView1.Columns[1].Width = 150;
for (int i = 0; i < dataGridView1.Columns.Count; i++)
{
if (dataGridView1.Columns[i] is DataGridViewImageColumn)
{
((DataGridViewImageColumn)dataGridView1.Columns[i]).ImageLayout =
DataGridViewImageCellLayout.Stretch;
```

```
}
}
}
catch(Exception ex)
{
con.Close();
MessageBox.Show(ex.Message);
}
}
private void mod_Click(object sender, EventArgs e)
{
con.Open();
int i =0;
SqlCommand cmd = new SqlCommand("UPDATE Employee SET Name = '" +
textname.Text + "',Phone = '" + textphone.Text + "',Email = '" + textemail.Text +
"',Address = '" + textaddress.Text + "',Image = @img WHERE idx = " +
dataGridView1.SelectedRows[0].Cells[0].Value + "",con);
MemoryStream stream = new MemoryStream();
pictureBox1.Image.Save(stream, System.Drawing.Imaging.ImageFormat.Bmp);
byte[] pic =stream.ToArray();
cmd.Parameters.AddWithValue("@img", pic);
i=cmd.ExecuteNonQuery();
if(i>0)
{
MessageBox.Show("수정완료 + i);
}
con.Close();
showdata();
clear();
}


private void del_Click(object sender, EventArgs e)
{
int i = 0;
con.Open();
SqlCommand cmd = new SqlCommand("DELETE FROM Employee WHERE idx = " +
dataGridView1.SelectedRows[0].Cells[0].Value + "", con);
i = cmd.ExecuteNonQuery();
con.Close();
showdata();
clear();
if (i > 0)
```

```
{
MessageBox.Show("삭제되었습니다);
}
}

private void save_Click(object sender, EventArgs e)
{
clear();
}

private void Main_Load(object sender, EventArgs e)
{
// TODO: 이코드는데이터를테이블에로드합니다필요한경우이코드를이동하거나제거할수있습니다
                this.employeeTableAdapter1.Fill(this.employeeDataSet1.Employee);
}

private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{

}

private void panel_Paint(object sender, PaintEventArgs e)
{

}

private void textsearch_TextChanged(object sender, EventArgs e)
{

}

private void label5_Click(object sender, EventArgs e)
{

}

private void button1_Click(object sender, EventArgs e)
{
Form1 ss = new Form1();
ss.Show();
this.Hide();
```

```
}

private void textaddress_TextChanged(object sender, EventArgs e)
{

}

private void textname_TextChanged(object sender, EventArgs e)
{

}

private void textemail_TextChanged(object sender, EventArgs e)
{

}

private void button2_Click(object sender, EventArgs e)
{
dataGridView1 = new DataGridView();
}

private void button2_Click_1(object sender, EventArgs e)
{
string sql = "SELECT idx, Image, Name, Phone, Email, Address FROM Employee
ORDER BY idx"
con.Open();
dataGridView1.RowTemplate.Height = 50;
dataGridView1.Height = 250;
DataTable dt = new DataTable();

command = new SqlCommand(sql, con);
da.SelectCommand = command;
dt.Clear();
da.Fill(dt);

dataGridView1.DataSource = dt;
dataGridView1.Columns[1].Width = 150;
for (int i = 0; i < dataGridView1.Columns.Count; i++)
{
if (dataGridView1.Columns[i] is DataGridViewImageColumn)
{
((DataGridViewImageColumn)dataGridView1.Columns[i]).ImageLayout =
```

```
DataGridViewImageCellLayout.Stretch;
}
}

con.Close();
}

private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
{
textname.Text = dataGridView1.SelectedRows[0].Cells[2].Value.ToString();
textphone.Text = dataGridView1.SelectedRows[0].Cells[3].Value.ToString();
textemail.Text = dataGridView1.SelectedRows[0].Cells[4].Value.ToString();
textaddress.Text = dataGridView1.SelectedRows[0].Cells[5].Value.ToString();
con.Open();
SqlCommand cmd = new SqlCommand("SELECT Image FROM Employee WHERE idx
= " + dataGridView1.SelectedRows[0].Cells[0].Value.ToString() + "", con);
da.SelectCommand = cmd;
DataSet ds = new DataSet();
byte[] mydata = new byte[0];
da.Fill(ds, "Employee");
DataRow myrow;
myrow = ds.Tables["Employee"].Rows[0];
mydata = (byte[])myrow["Image"];
MemoryStream stream = new MemoryStream(mydata);
pictureBox1.Image = Image.FromStream(stream);
con.Close();
}

private void show_Click_1(object sender, EventArgs e)
{
showdata();
}

private void search_Click(object sender, EventArgs e)
{
con.Open();
SqlCommand cmd = con.CreateCommand();
cmd.CommandType = CommandType.Text;
if (comboBox1.Text == "Name")
{
cmd.CommandText = "SELECT * FROM Employee WHERE Name like('" +
textsearch.Text + "%')"
cmd.ExecuteNonQuery();
```

```
DataTable dt = new DataTable();
SqlDataAdapter da = new SqlDataAdapter(cmd);
da.Fill(dt);
dataGridView1.DataSource = dt;
}
else if (comboBox1.Text == "Phone")
{
cmd.CommandText = "SELECT * FROM Employee WHERE Phone like('" +
textsearch.Text + "%')"
cmd.ExecuteNonQuery();
DataTable dt = new DataTable();
SqlDataAdapter da = new SqlDataAdapter(cmd);
da.Fill(dt);
dataGridView1.DataSource = dt;
}
else if (comboBox1.Text == "Email")
{
cmd.CommandText = "SELECT * FROM Employee WHERE Email like('" +
textsearch.Text + "%')"
cmd.ExecuteNonQuery();
DataTable dt = new DataTable();
SqlDataAdapter da = new SqlDataAdapter(cmd);
da.Fill(dt);
dataGridView1.DataSource = dt;
}
else if (comboBox1.Text == "Address")
{
cmd.CommandText = "SELECT * FROM Employee WHERE Address like('" +
textsearch.Text + "%')"
cmd.ExecuteNonQuery();
DataTable dt = new DataTable();
SqlDataAdapter da = new SqlDataAdapter(cmd);
da.Fill(dt);
dataGridView1.DataSource = dt;

}
else
                {
MessageBox.Show("검색메뉴를선택하세요);
}
con.Close();
}
}
```

```
}
```

AES 알고리즘

```
using System;
using System.Collections.Generic;
using System.Text;
using System.IO;
using System.Security.Cryptography;

namespace Block
{
class AES256Cipher
    {
public String AES_encrypt(String Input, String key)
{
RijndaelManaged aes = new RijndaelManaged();
aes.KeySize = 256;
aes.BlockSize = 128;
aes.Mode = CipherMode.CBC;
aes.Padding = PaddingMode.PKCS7;
aes.Key = Encoding.UTF8.GetBytes(key);
aes.IV = new byte[] { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };

var encrypt = aes.CreateEncryptor(aes.Key, aes.IV);
byte[] xBuff = null
using (var ms = new MemoryStream())
{
using (var cs = new CryptoStream(ms, encrypt, CryptoStreamMode.Write))
{
byte[] xXml = Encoding.UTF8.GetBytes(Input);
cs.Write(xXml, 0, xXml.Length);
}

xBuff = ms.ToArray();
}

String Output = Convert.ToBase64String(xBuff);
return Output;
}
```

```
public String AES_decrypt(String Input, String key)
{
RijndaelManaged aes = new RijndaelManaged();
aes.KeySize = 256;
aes.BlockSize = 128;
aes.Mode = CipherMode.CBC;
aes.Padding = PaddingMode.PKCS7;
aes.Key = Encoding.UTF8.GetBytes(key);
aes.IV = new byte[] { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };

var decrypt = aes.CreateDecryptor();
byte[] xBuff = null
using (var ms = new MemoryStream())
{
using (var cs = new CryptoStream(ms, decrypt, CryptoStreamMode.Write))
{
byte[] xXml = Convert.FromBase64String(Input);
cs.Write(xXml, 0, xXml.Length);
}

xBuff = ms.ToArray();
}

String Output = Encoding.UTF8.GetString(xBuff);
return Output;
}
}
}
```

5. 결론

본 프로젝트는 인터넷 기술이 발전함과 동시에 급증하고 있는 이미지 도용 및 저작권 침해 예방을 목적으로 하며 워터마킹, 스테가노그래피 시스템 기법의 이해와 응용프로그램 구현을 위하여 시작 하였다.

이 프로젝트는 AES암호 알고리즘을 사용한 데이터 은닉 기법을 활용하였고, 워터마킹의 정보 삽입과 데이터 베이스의 연동으로 사용자가 편리하게 개인정보를 관리하며 수정에 용이할 수 있도록 구현하였다.

프로젝트를 수행하며 스테가노그래피가 수학과 복잡합 알고리즘에 기반한 암호화 기술뿐만이 아니라는 것을 배웠고, 이미지 파일의 구조에 대한 이해와 조작으로 다른 파일이나 정보를 어떻게 은닉하고, 변경하는지 이해 할 수 있었다.

6. 발표 PPT


졸업작품 최종 보고서
이미지 워터 마킹
2016 . 5 . 17
중부대학교 정보보호학과
지도교수님 : 유승재 교수님
5조 Block's
Information security

목차

Chapter 1

조원 소개 및 역할

01
조원소개

조원 소개 및 역할

1 김태윤
- C#, 작품 총괄 및 프로그램개발
- MS sql

2 신용하
- C# ,Windows forms
- 자료수집, 보고서 작성

3 최동규
- C# ,워터마크 삽입
- 자료수집 , 보고서 작성

4 김민수
- C# ,DB 연동
- 자료수집 , 보고서 작성

Chapter 2

주제 선정 이유

02
주제 선정 이유

이러한 피해 사건 사고들

AOA 민아, 성인 사이트에 버젓이 사진 도용

이거 지금 네이버에 원빈아버지 젊은 시절이라고 기사 떴는데 이거 저희 아버지 해병대 사진 이거든요.. 이거 한밤의 TV연예라는곳에도 나왔다는데.... 저희 아빠 사진인데 이거 어떻게 해야하나요? 아시는분 도와주세요

원빈 아버지 젊은시절..."유전자의 중요성 다시보기"

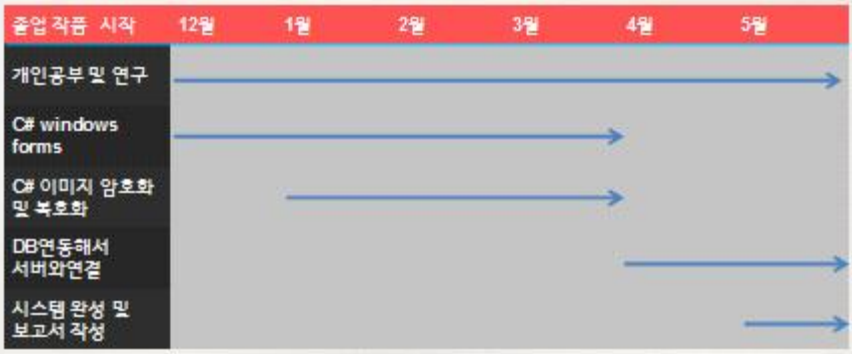
요즘 들어 사진도용 사건이 많아져서 수많은 피해가 발생하고 있다.

Chapter 3

추진 경과

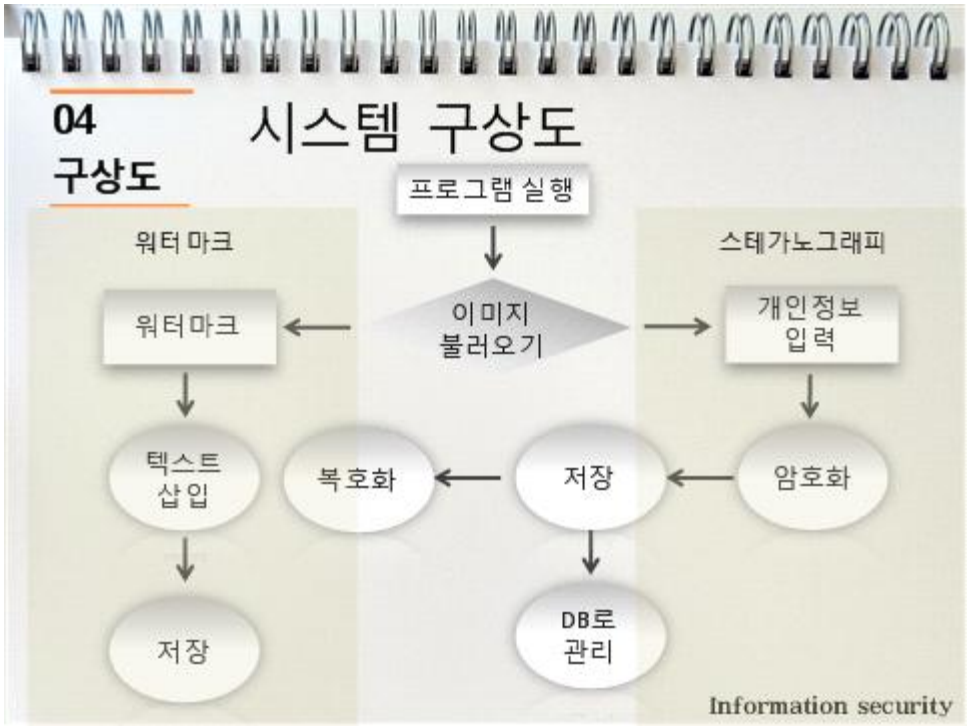
03
추진 경과

졸업작품 추진 경과



Chapter 4

졸업 작품 구상도


04
구상도
시스템 구상도
프로그램 실행
워터 마크
스테가노그래피
워터마크
이미지
불러오기
개인정보
입력
텍스트
삽입
복호화
저장
암호화
저장
DB로
관리
Information security

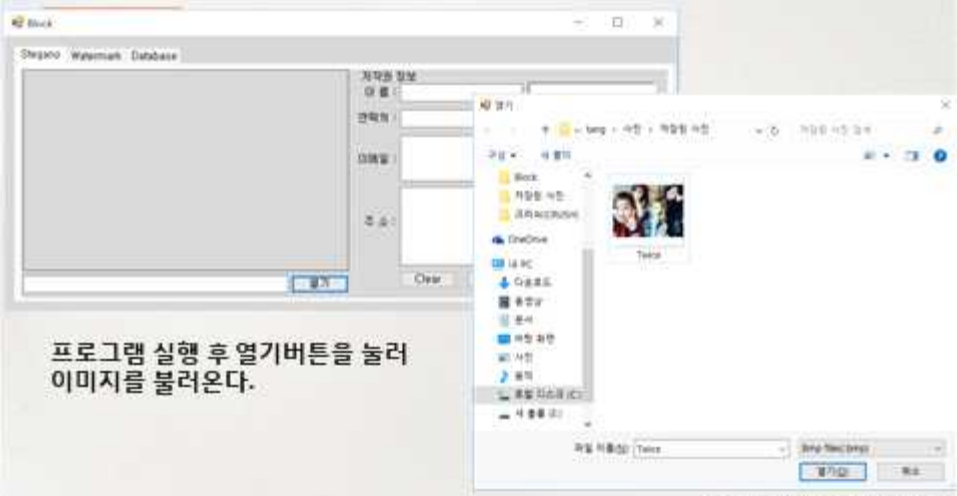

04
구상도
시스템 구상도
프로그램
실행
개인 정보 입력
암호화 할 이미지 파일 열기
암호화
데이터 베이스
워터마크 삽입
워터마크 입힐 이미지 열기
저장
관리
관리자
사용자 정보
Information security

Chapter 5

프로그램 개발 및 운영

05 프로그램 개발 및 운영

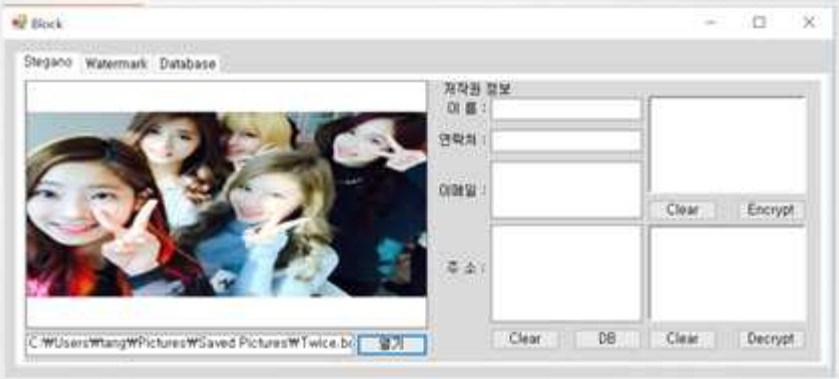
프로그램



프로그램 실행 후 열기버튼을 눌러 이미지를 불러온다.

05 프로그램 개발 및 운영

프로그램



개인정보를 삽입 후 암호화 버튼을 누르고 저장한다.

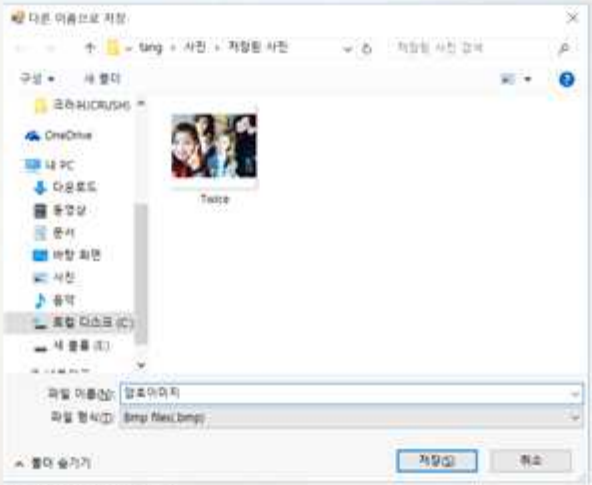
05 프로그램 개발 및 운영

프로그램

암호화 된 이미지를 저장합니다.

05 프로그램

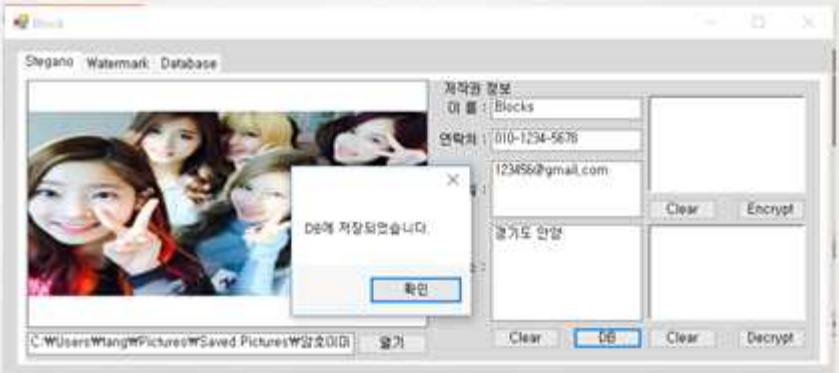
프로그램 개발 및 운영



암호화 된 이미지를 저장합니다.

05 프로그램

프로그램 개발 및 운영



암호화 된 이미지 저장 후 DB버튼을 누르게 되면 DB에 저장 하게 됩니다.

05 프로그램 개발 및 운영

프로그램

이미지를 다시 불러온 후 복호화 버튼을 누르면 암호화된 정보를 불러온다.

05 프로그램 개발 및 운영

프로그램

암호화된 정보 확인 후 저작권 상자에 들어가는 것을 확인할 수 있다

05 프로그램 개발 및 운영

프로그램

두번째 탭은 워터마크 탭입니다.

05 프로그램 개발 및 운영

프로그램

이미지를 불러온 후 워터마크 할 텍스트를 입력하고 옵션을 설정 합니다.

05 프로그램 개발 및 운영

프로그램

텍스트 입력 후 위치, 투명도 선택 후 미리보기를 클릭하면 위와 같이 나옵니다.

05 프로그램 개발 및 운영

프로그램

워터마크를 입력하고 저장 후 확인을 하면 이렇게 워터마크가 된 이미지가 보입니다.

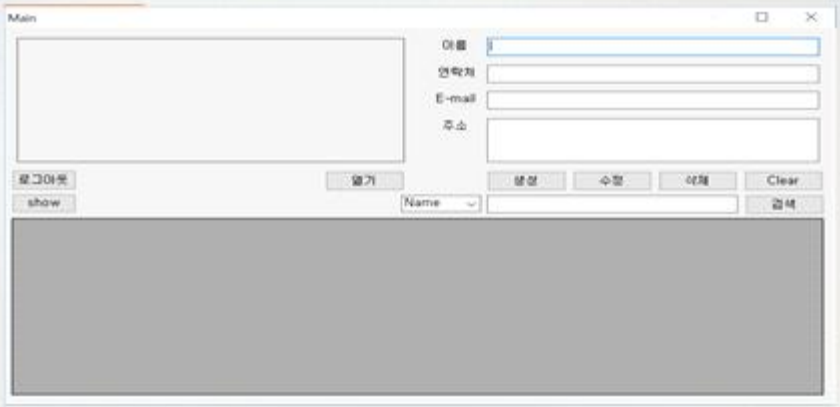
05 프로그램 개발 및 운영

프로그램

세번째 탭에 데이터베이스 라고 되어있는 곳에 관리자 ID로 들어가게 됩니다.

05 프로그램 개발 및 운영

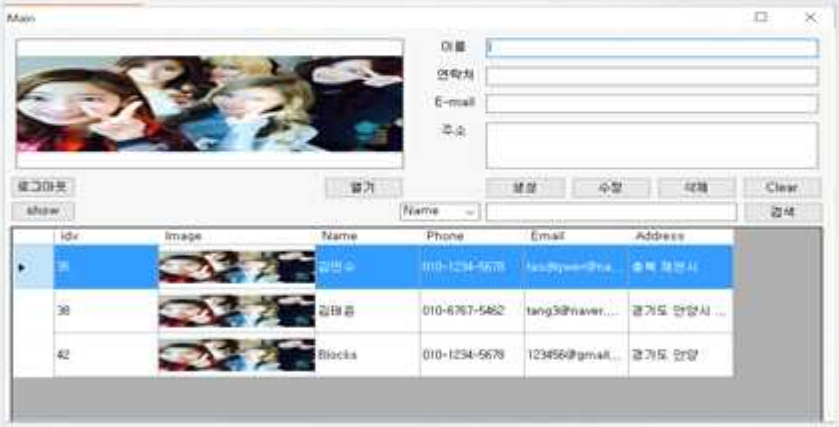
프로그램



처음에 저장한 이미지와 개인정보를 불러 올 수 있고 새롭게 생성,수정도 할 수 있다.

05 프로그램 개발 및 운영

프로그램



DB내에서 Select, Insert, Update, Delete 할 수 있다.

05 프로그램 개발 및 운영

프로그램

DB내에서 검색창에 이름, 번호, 메일, 주소를 검색해서 찾을수 있다.

Chapter 6

결론 및 기대효과

06
결론

결론 및 기대효과

자기 자신의 사진도용을 예방하기 위해 프로그램을 만들었다.

- 프로그램을 실행시킨 후 이미지에 파일을 덮어씌워 워터마크 기법으로 도용을 방지한다.
- 저작권자가 개인정보를 입력했을 때 개인정보를 DB에서 관리한다.
- 프로그램으로 개인정보를 삽입했을 때 저작권자가 나중에 잊어버릴수도 있으므로 DB에서 관리해주며 찾을 때 편리하게 만들었다.


감사합니다
Information security