**컴퓨터보안 실습과제 : Steganography**

2021.3.22

12150981

박중규

**1. 인코딩 아이디어**

* 핵심아이디어 : 픽셀데이터의 색상값은 1정도의 차이가 나도 육안으로 알아볼 수 없기때문에, 각 픽셀데이터의 최하단 비트에 0 또는 1의 값을 숨기도록 하였습니다. 또 파일의 크기에도 영향을 미치지 않게 됩니다.
* BMP파일데이터의 46~53번째는 헤더의 끝부분으로써 앞 4바이트는 색상 테이블에서 실제 사용되는 색상 수를 나타내고 뒤 4바이트는 비트맵을 표현하기 위해 필요한 색상 인덱스 수를 나타내는데, 이 8바이트 공간에 사용자에게 입력받은 메시지의 크기를 숨겼습니다.   
  1바이트에 최대 255까지의 숫자를 담을 수 있으므로, 8바이트니까 최대 255 \* 8 = 2040 글자수만큼 메시지를 입력받을 수 있도록 설계하였습니다.
* gets\_s를 사용하여 사용자에게 띄어쓰기가 포함된 문자열을 입력받았고,  
  각 문자를 8비트로 나눠서 ( 메시지 크기 ) \* 8크기의 2차원행렬에 이진수로 저장하였습니다.
* 54번째부터 픽셀데이터의 시작이므로,  
  픽셀데이터 1바이트당 0또는1의 값을 숨기도록 하였습니다.  
  이 때 픽셀데이터에 ‘%’ 연산을 사용하여 저장하려는 비트의 값과 같으면 그대로 파일데이터를 옮겼고, 다르면 1을 빼거나 추가하여 데이터를 옮겼습니다. (픽셀데이터의 값이 0일 경우에만 1을 더해주고 나머지 경우에는 1을 빼주었습니다.)
* 1바이트에 1비트씩 숨길 수 있으므로, 8바이트 당 문자 1개의 정보를 저장하게 됩니다.
* 메시지를 숨긴 이후에는 나머지 파일데이터는 똑같이 옮긴 이후,  
  최종적으로 메시지를 숨긴 “stego.bmp” 파일을 작성하였습니다.

**2. 디코딩 아이디어**

* 인코딩의 방식의 역순으로 진행하게 됩니다.
* 먼저 “stego.bmp” 파일데이터를 배열에 담은 이후,  
  파일포인터를 46번째 바이트로 옮겨서 46~53번째 바이트에 저장된 수를 모두 계산하여 숨겨진 메시지의 크기를 구합니다.
* 이후 인코딩 때와 같이 이진수를 담을 ( 메시지 크기) \* 8 크기의 2차원 배열을 만들고,  
  54번째 바이트부터 메시지 크기만큼 비트를 옮깁니다.  
  ‘%’ 연산을 사용하게 되면 원하는 값을 얻어올 수 있게 됩니다.
* 값을 다 담은 이후 8비트의 값을 모두 더하여 정수로 만들어주고, 정수를 아스키코드에 의해 문자로 변환시켜 1개의 문자로 만들어줍니다.  
  그 문자들을 모두 더하여 하나의 문자열을 만들고, 콘솔에 출력하게 됩니다.

**3. 샘플 이미지**

