블록암호 PIPO를 이용한 이미지 암호화

1871005 강예준

https://youtu.be/hXNq3474tgA

한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB
https://crypto.modoo.at

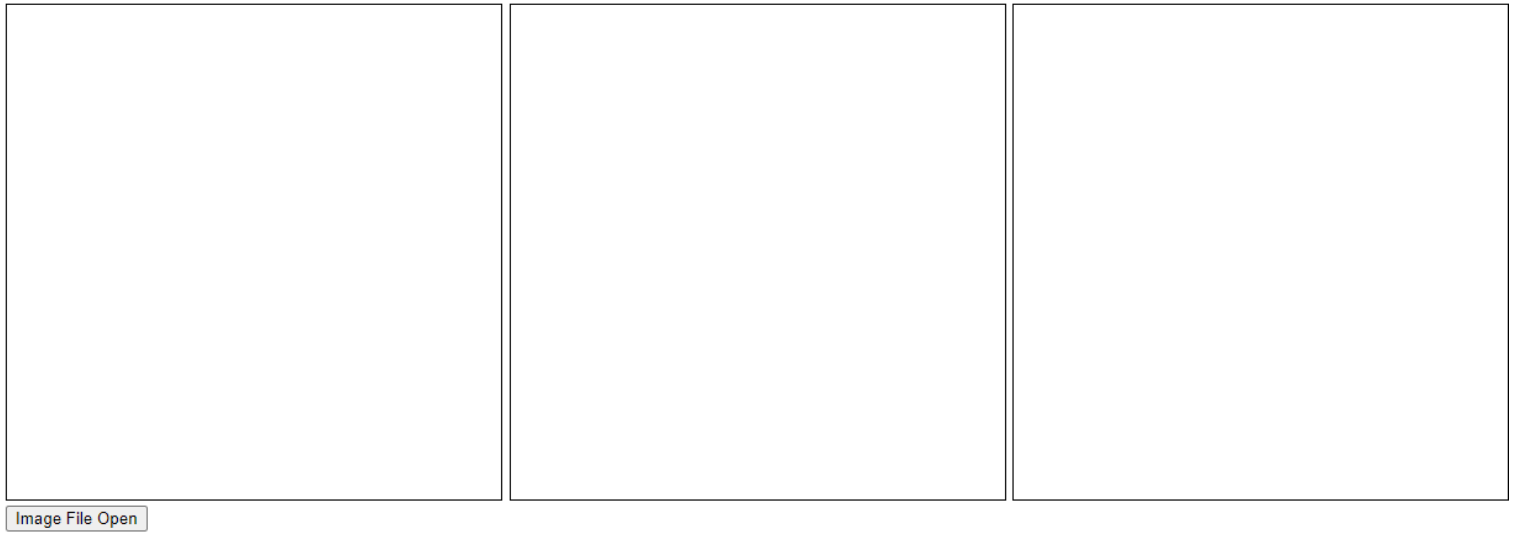
Contents

CryptoCraft LAB

1. 코드 설명

- 블록암호 PIPO를 자바스크립트를 통해 구현

```
<body onload="draw();">
    <canvas id="canvas" width=400 height=400 style=' border : 1px solid #000 ' ;></canvas>
    <canvas id="canvasENC" width=400 height=400 style=' border : 1px solid #000 ' ;></canvas>
    <canvas id="canvasDEC" width=400 height=400 style=' border : 1px solid #000 ' ;></canvas>

    <br>
    <button onclick="openTextFile()">Image File Open</button>
</body>
```



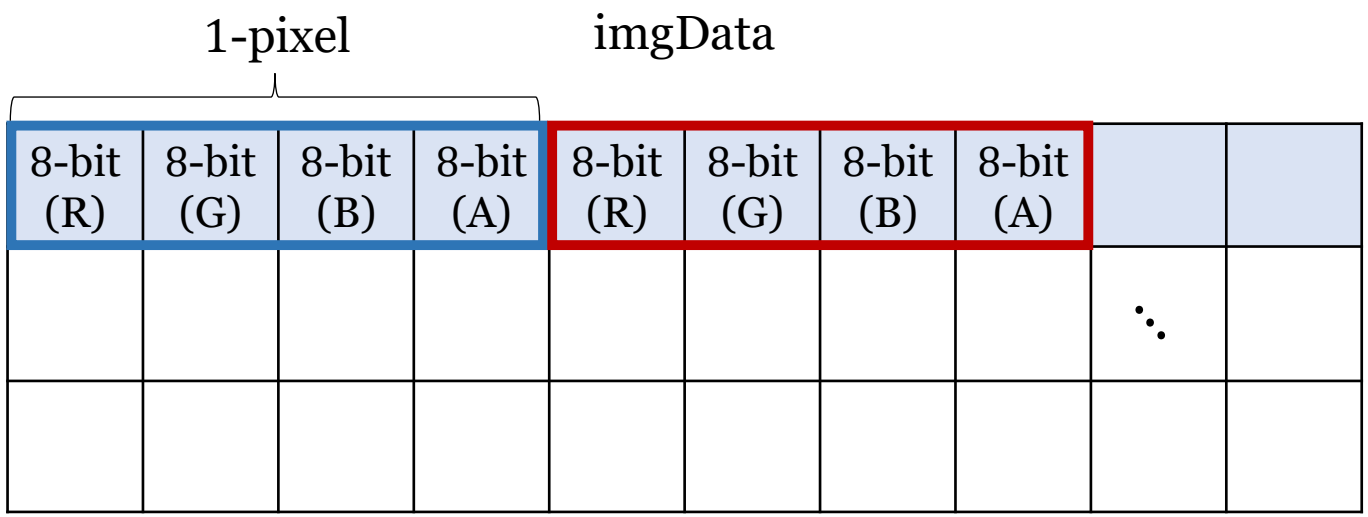

```
//파일 업로드
function openTextFile() {
    var input = document.createElement("input");    //업로드 할 수 있게 해줌

    input.type = "file";
    input.accept = "image/*";   //이미지 파일 받을 수 있게
    input.id = "uploadInput";

    input.click();
    input.onchange = function (event) {
        processFile(event.target.files[0]);
    };
}

function processFile(file) {
    var reader = new FileReader();

    reader.onload = function () {
        var result = reader.result;
        img.src = result;
    };
    reader.readAsDataURL(file);
}
```

Base64 : A-Z, a-z, 0-9, /,+

1. 코드 설명

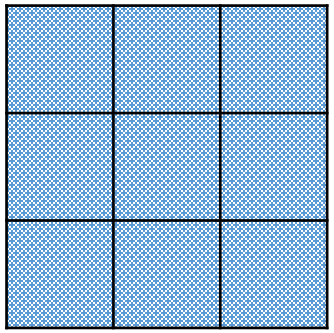
- draw 함수

```
function draw() {
    var canvas = document.getElementById("canvas");            //canvas 객체 받아온다.
    var canvasENC = document.getElementById('canvasENC');   //canvasENC 객체 받아온다.
    var canvasDEC = document.getElementById('canvasDEC');  //canvasDEC 객체 받아온다.
    var ctx = canvas.getContext('2d');
    var ENC_ctx = canvasENC.getContext('2d');

    img.onload = function () {
        img_size = 640000;
        //길이가 원래는 160,000(400*400)개인데 각 픽셀별로 4개(RGB+투명도)가 있기 때문에 160,000*4 = 640,000개가 된다.
        ctx.drawImage(img, 0, 0, 400, 400); //이미지를 그린다.

        var imageData = ctx.getImageData(0, 0, 400, 400); //(0,0)부터 400*400만큼의 이미지 데이터를 가져와
        ENCPixels(imageData.data); // 이미지 암호화
        drawENCImage(imageData);   // 이미지 출력

        DECPixels(imageData.data);
        drawDECImage(imageData);
    };
}
```



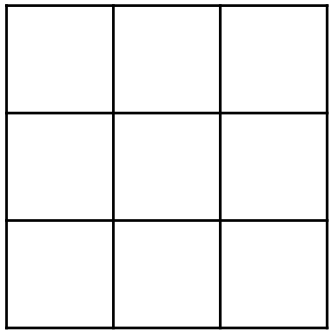
JS_PIPO의 평문으로 2-pixel씩 입력

Encrypt (imgData)



imgData (Encrypted)

Decrypt (imgData)



imgData (Decrypted)

1. 코드 설명

- ENCPixels 함수

```
function ENCPixels(imgData) {
    RND_KEY = create2DArray(img_size/8, 112);

    for (var i = 0, r = 0; i < img_size; i += 8, r++) {

        PLAIN_GEN(imgData[i + 0], imgData[i + 1], imgData[i + 2], imgData[i + 3],
            imgData[i + 4], imgData[i + 5], imgData[i + 6], imgData[i + 7]);

        tmp = ROUND_KEY_GEN();  //라운드키가 있어야 복호화 가능 (키생성)

        for (var j = 0; j < 112; j++) {
            RND_KEY[r][j] = tmp[j];
        }
        PLAIN_TEXT = ENC(PLAIN_TEXT, RND_KEY[r], CIPHER_TEXT);

        imgData[i + 0] = PLAIN_TEXT[3];  //RED
        imgData[i + 1] = PLAIN_TEXT[2];  //GREEN
        imgData[i + 2] = PLAIN_TEXT[1];  //BLUE
        imgData[i + 3] = PLAIN_TEXT[0];  //투명도

        imgData[i + 4] = PLAIN_TEXT[7]; //RED
        imgData[i + 5] = PLAIN_TEXT[6]; //GREEN
        imgData[i + 6] = PLAIN_TEXT[5]; //BLUE
        imgData[i + 7] = PLAIN_TEXT[4]; //투명도
    }
}
```

Q & A