**AES Encryption**

Nama Anggota:

1. Arief Saferman (1806148656)
2. Kevin Darmawan (1806148744)
3. Ramadhan Kalih Sewu (1806148826)
4. Yogie Wisesa (1806148851)

```
function [res] = KeySchedule(key)
    rcon = {'01' '02' '04' '08' '10' '20' '40' '80' '1b' '36';
            '00' '00' '00' '00' '00' '00' '00' '00' '00' '00';
            '00' '00' '00' '00' '00' '00' '00' '00' '00' '00';
            '00' '00' '00' '00' '00' '00' '00' '00' '00' '00'};
    collen = 4;
    keyRnd = [];
    rcon = reshape(hex2dec(rcon), size(rcon));
    res = { key };
    for round = 1:10
        prevKey = cell2mat(res(round));
        rot = RotWord(prevKey(:,collen));
        a = prevKey(:,1);
        b = SubBytes(rot);
        c = rcon(:,round);
        keyRnd(:,1) = bitxor( bitxor(a,b), c );
        for col = 2:collen
            a = prevKey(:,col);
            b = keyRnd(:,col-1);
            keyRnd(:,col) = bitxor( a,b );
        end
        res{end+1} = keyRnd;
    end
end
```

Fungsi schedule digunakan untuk menghasilkan key untuk masing-masing round. Berdasarkan video animasi yang ada total akan 11 key yang terbagi oleh:

- 9 key untuk main rounds,
- 1 key sebagai initial round sebelum masuk ke proses enkripsi, dan
- 1 key terakhir untuk final round.

Di fungsi ini kita akan mengambil rotWord yaitu kolom terakhir dari sebuah state matrix lalu kita rotate rotword tersebut dari baris ke-1 ke baris terakhir. Lalu, kita substitute dengan S-BOX yang sudah kita definisikan. Setelah itu kita lakukan bitxor antara kolom pertama state matrix dengan rotword yang sudah di substitute lalu bitxor lagi dengan kolom konstanta R kolom bersesuaian dengan urutan rondenya. Misal ronde pertama maka kita akan mengambil kolom pertama dari matrix Rcon. Hasilkan lalu akan beruba matrix 4x1 yang akan di letakan di kolom pertama untuk matrix key di round pertama tersebut. Lalu untuk kolom ke-2 sampai ke-4 lakukan bitxor dengan state matrix kolom ke 2 dengan previous matrix yang sudah diletakan di roundkey.

```
function [state] = AES(state, key)
    % get every round key
    keyRnd = KeySchedule(key);
    % initial round
    state = bitxor(state, cell2mat(keyRnd(:,1)));
    % 10 round
    for round = 1:10
        % Byte Substitution
        state = SubBytes(state);
        % Shift Rows
        state = ShiftRows(state);
        % Mix Columns
        if (round ~= 10)
            state = MixColumns(state);
        end
        % XOR with Round Key
        state = bitxor(state, cell2mat(keyRnd(:,round+1)));
    end
end
```

Fungsi utama dari program ini ada pada AES.m dimana alur dari algoritma AES diimplementasikan. Algoritma ini melibatkan proses key schedule yang mengextend cipher key menjadi 11 key. Proses **ronde** dalam enkripsi melibatkan operasi SubBytes, ShiftRows, MixColumns dan pada akhir round akan di bit xor dengan round key dari key schedule. Pada masing-masing ronde, berikut yang dilakukan:

- Ronde inisial, state (teks) akan di bitxor dengan round key 1
- Ronde 1-9, dari output ronde sebelumnya – substitusi byte – shift rows – mix columns
- Ronde 10, dari output ronde sebelumnya – substitusi byte – shift rows – bit xor dengan round key 10

```
function [newstate] = MixColumns(state)
    % Rijndael's Galois Field
    newstate = zeros(size(state));
    RGF = [2 3 1 1; 1 2 3 1; 1 1 2 3; 3 1 1 2];
    for row = 1:4
        for col = 1:4
            gal = RGF(row,:);
            val = zeros(1,4);
            for e = 1:4
                val(e) = state(e,col);
                if (gal(e) >= 2)
                    val(e) = 2 * val(e);
                    if (val(e) > 255)
                        val(e) = bitxor(val(e), hex2dec('1B'));
                    end
                    if (gal(e) == 3)
                        val(e) = bitxor(val(e), state(e,col));
                    end
                    if (val(e) > 255)
                       val(e) = bitand(val(e), 255);
                    end
                end
            end
            newstate(row,col) = bitxor(bitxor(bitxor(val(1), val(2)), val(3)),
val(4));
        end
    end
end
```

Fungsi MixedColumns melakukan operasi modulo multiplication untuk matrix Rijndael's Galois Field (RGF) terhadap masing masing kolom pada setiap round dan mensubstitusi elemen tsb dengan hasilnya.

Referensi: https://www.youtube.com/watch?v=Tx_37dF03ig

```
function [out] = RotWord(in)
    out = circshift(in, -1);
end
```

Fungsi RotWord berfungsi untuk melakukan rotasi tiap word (kolom) dalam proses Key Scheduler.

- Row 1 – tidak di rotasi
- Row 2 – rotasi 1 byte ke kiri
- Row 3 – rotasi 2 byte ke kiri
- Row 4 – rotasi 3 byte ke kiri

```
function [state] = ShiftRows(state)
    for i = 2:4
        state(i, :) = circshift(state(i, :), [0, 1-i]);
    end
end
```

Fungsi ShiftRows me-rotate byte pada state untuk masing masing row sebanyak row-1

```
text = {'32' '88' '31' 'e0';
        '43' '5a' '31' '37';
        'f6' '30' '98' '07';
        'a8' '8d' 'a2' '34'}

ckey = {'2b' '28' 'ab' '09';
        '7e' 'ae' 'f7' 'cf';
        '15' 'd2' '15' '4f';
        '16' 'a6' '88' '3c'}

text = reshape(hex2dec(text), size(text));
ckey = reshape(hex2dec(ckey), size(ckey));

res = AES(text, ckey);
display(res);
```

Pada file source berikut terdapat matrix input (plaintext yang diubah ke matrix hexa) dan cipher key (dalam matrix hexa), untuk memudahkan pemrosesan data, kedua matrix diubah menjadi matrix desimal berdimensi 4x4. File source ini juga berfungsi sebagai main function yang menerima input dan menunjukkan hasil.

```
%% *SubBytes()* Transformation
% Transformation in the Cipher that processes the State using a nonlinear
% byte substitution table (S-box) that operates on each of the State bytes
% independently

function [res] = SubBytes(state)
    sbox = {'63' '7c' '77' '7b' 'f2' '6b' '6f' 'c5' '30' '01' '67' '2b' 'fe' 'd7' 'ab' '76';
            'ca' '82' 'c9' '7d' 'fa' '59' '47' 'f0' 'ad' 'd4' 'a2' 'af' '9c' 'a4' '72' 'c0';
            'b7' 'fd' '93' '26' '36' '3f' 'f7' 'cc' '34' 'a5' 'e5' 'f1' '71' 'd8' '31' '15';
            '04' 'c7' '23' 'c3' '18' '96' '05' '9a' '07' '12' '80' 'e2' 'eb' '27' 'b2' '75';
            '09' '83' '2c' '1a' '1b' '6e' '5a' 'a0' '52' '3b' 'd6' 'b3' '29' 'e3' '2f' '84';
            '53' 'd1' '00' 'ed' '20' 'fc' 'b1' '5b' '6a' 'cb' 'be' '39' '4a' '4c' '58' 'cf';
            'd0' 'ef' 'aa' 'fb' '43' '4d' '33' '85' '45' 'f9' '02' '7f' '50' '3c' '9f' 'a8';
            '51' 'a3' '40' '8f' '92' '9d' '38' 'f5' 'bc' 'b6' 'da' '21' '10' 'ff' 'f3' 'd2';
            'cd' '0c' '13' 'ec' '5f' '97' '44' '17' 'c4' 'a7' '7e' '3d' '64' '5d' '19' '73';
            '60' '81' '4f' 'dc' '22' '2a' '90' '88' '46' 'ee' 'b8' '14' 'de' '5e' '0b' 'db';
            'e0' '32' '3a' '0a' '49' '06' '24' '5c' 'c2' 'd3' 'ac' '62' '91' '95' 'e4' '79';
            'e7' 'c8' '37' '6d' '8d' 'd5' '4e' 'a9' '6c' '56' 'f4' 'ea' '65' '7a' 'ae' '08';
            'ba' '78' '25' '2e' '1c' 'a6' 'b4' 'c6' 'e8' 'dd' '74' '1f' '4b' 'bd' '8b' '8a';
            '70' '3e' 'b5' '66' '48' '03' 'f6' '0e' '61' '35' '57' 'b9' '86' 'c1' '1d' '9e';
            'e1' 'f8' '98' '11' '69' 'd9' '8e' '94' '9b' '1e' '87' 'e9' 'ce' '55' '28' 'df';
            '8c' 'a1' '89' '0d' 'bf' 'e6' '42' '68' '41' '99' '2d' '0f' 'b0' '54' 'bb' '16'};
    res = arrayfun( @(x) sbox(floor(x/16) + 1, mod(x, 16) + 1),state );
    res = reshape(hex2dec(res), size(state));
end
```

Fungsi SubBytes.m melakukan substitusi masing-masing element state terhadap matriks byte transformation S-box.

Output akan dalam bentuk Decimal:

Hasilnya Sesuai dengan yang diharapkan

```
text =

    '32'    '88'    '31'    'e0'
    '43'    '5a'    '31'    '37'
    'f6'    '30'    '98'    '07'
    'a8'    '8d'    'a2'    '34'


ckey =

    '2b'    '28'    'ab'    '09'
    '7e'    'ae'    'f7'    'cf'
    '15'    'd2'    '15'    '4f'
    '16'    'a6'    '88'    '3c'


res =

    57     2   220    25
    37   220    17   106
   132     9   133    11
    29   251   151    50
```