컴퓨터보안 7 주차 RC4 과제 / 2018920065 루안리치

프로그래밍 언어: C

소스코드: (실행결과 -> p.5)

```
//
//  main.c
//  RC4_2018920065
//
//  Created by LUAN LI CHI on 2022/04/27.
//

#include <stdio.h>
#include <stdlib.h>

int S[8]={0,1,2,3,4,5,6,7};
int T[8];
int K[4] = {1,2,3,6};
int P[4] = {1,2,2,2};
int binK[4];
int binP[4];
int C[4];
int P16[16];
int K16[16];
int C16[16];


void printarray(int arr[], int size){
    printf(" { ");
    for(int i=0;i<size;i++){
        printf("%d ",arr[i]);
    }
    printf("}\n");
}

int bintodec(int b[]){
    int i, d=0, x=8;
```

```
    for(i=0;i<4;i++){
        if(b[i]==1){
            d+=x;
        }
        x=x/2;
    }
    return d;
}

void dectobinK(int d){
    int x=8, i=0;
    while(x>0){
        if(d>=x){
            binK[i] = 1;
            d=d-x;
        }
        else{
            binK[i] = 0;
        }
        i++;
        x=x/2;
    }
}

void dectobinP(int d){
    int x=8, i=0;
    while(x>0){
        if(d>=x){
            binP[i] = 1;
            d=d-x;
        }
        else{
            binP[i] = 0;
        }
        i++;
        x=x/2;
    }
}
```

```
void initialT(){
	for(int i=0;i<8;i++){
		if(i>3){
			T[i]=K[i-4];
		}
		else{
			T[i]=K[i];
		}
	}
	printf("initial T:");
	printarray(T, 8);
}

void initialPermutation(){
	int i, temp;
	int j=0;
	printf("-\ninitial permutation:\n");
	for(i=0;i<8;i++){
		j=(j+S[i]+T[i])%8;
		temp=S[i];
		S[i]=S[j];
		S[j]=temp;
		printf("-\nround %d: j=%d\n",i, j);
		printf("S[%d]:",i);
		printarray(S, 8);
	}
}

int xorcalculate(int k, int p){
	int out;
	int temp[4];
	dectobinK(k);
	dectobinP(p);
	//printarray(binK, 4);
	//printarray(binP, 4);
	for(int i=0;i<4;i++){
		if(binK[i] == binP[i]){
```

```
            temp[i]=0;
        }
        else{
            temp[i]=1;
        }
    }
    out=bintodec(temp);
    //printarray(temp, 4);
    //printf("%d & %d, out %d\n",k, p, out);
    return out;
}


void generationLoop(){
    int i=0;
    int j=0;
    int temp;
    int t,k;
    printf("-\ngeneration loop:\n");
    for(int n=0;n<4;n++){
        i=(i+1)%8;
        j=(j+S[i])%8;
        temp=S[i];
        S[i]=S[j];
        S[j]=temp;
        printf("-\nround %d: i=%d, j=%d\nS[%d]:",n, i, j, n);
        printarray(S, 8);
        t=(S[i]+S[j])%8;
        k=S[t];
        C[n]=xorcalculate(k,P[n]);
        printf("C[%d]=%d\n",n, C[n]);
    }
    printf("-\nC:");
    printarray(C, 4);
}


int main(int argc, const char * argv[]) {
```

```
    initialT();
    initialPermutation();
    generationLoop();

    return 0;
}
```

실행결과:

```
initial T: { 1 2 3 6 1 2 3 6 }
-
initial permutation:
-
round 0: j=1
S[0]: { 1 0 2 3 4 5 6 7 }
-
round 1: j=3
S[1]: { 1 3 2 0 4 5 6 7 }
-
round 2: j=0
S[2]: { 2 3 1 0 4 5 6 7 }
-
round 3: j=6
S[3]: { 2 3 1 6 4 5 0 7 }
-
round 4: j=3
S[4]: { 2 3 1 4 6 5 0 7 }
-
round 5: j=2
S[5]: { 2 3 5 4 6 1 0 7 }
-
round 6: j=5
S[6]: { 2 3 5 4 6 0 1 7 }
-
round 7: j=2
S[7]: { 2 3 7 4 6 0 1 5 }
-
```

```
-
generation loop:
-
round 0: i=1, j=3
S[0]: { 2 4 7 3 6 0 1 5 }
C[0]=4
-
round 1: i=2, j=2
S[1]: { 2 4 7 3 6 0 1 5 }
C[1]=3
-
round 2: i=3, j=5
S[2]: { 2 4 7 0 6 3 1 5 }
C[2]=2
-
round 3: i=4, j=3
S[3]: { 2 4 7 6 0 3 1 5 }
C[3]=3
-
C: { 4 3 2 3 }
Program ended with exit code: 0
```