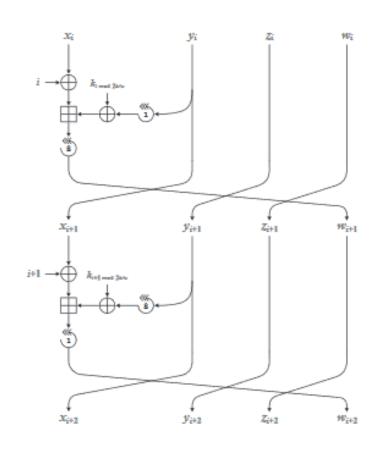
AVR 프로그래밍

7강 정보컴퓨터공학과 권혁동





- 지금까지 배운 기법을 기반으로 CHAM을 구현
- CHAM-64/128을 구현
- ARX 구조 알고리즘
- 88라운드를 반복
- 다음 테스트 벡터를 사용
 - RK: 0x0301, 0x0705, 0x0b09, 0x0f0d, 0x1311, 0x1715, 0x1b19, 0x1f1d, 0x151e, 0x0308, 0x3932, 0x2f24, 0x4d46, 0x5b50, 0x616a, 0x777c
 - PT: 0x1100, 0x3322, 0x5544, 0x7766
 - CT: 0x6579 0x1204 0x123f 0xe5a9





u16 rk[16] __attribute_ ((section(".RKSection"))) = {0x0301, 0x0705, 0x0b09, 0x0f0d, 0x1311, 0x1715, 0x1b19, 0x1f1d, 0x151e, 0x0308, 0x3932, 0x2f24, 0x4d46, 0x5b50, 0x616a, 0x777c}; $u16 pt[4] = {0x1100, 0x3322, 0x5544, 0x7766};$ ■ Market AVR/GNU Common // CT = 0x6579 0x1204 0x123f 0xe5a9 General <segmentname>=<address>, for example .boot=0xff Segment values syntax: ☑ Output Files ■ Martin Avraga Avr The AVR is a Harvard architecture CPU. This means that it separates instruction memory and AVR port of GCC: General data memory. The gcc was originally designed to support Von Neumann architectures which Preprocessor define a single storage structure to hold both instructions and data. This dichotomy is solved by a series of nifty tricks in the AVR port of gcc, of which three should be noted: ☑ Directories Optimization The .text segment starts at 0x0. Debugging Warnings FLASH segment Miscellaneous ■ Martin AVR/GNU Linker General Libraries ☑ Memory Settings AVR/GNU Assembler ☑ General Debugging RKSection=0x0400 ■ Marchiver Archiver ☑ General EEPROM segment

- 변수 선언
 - 메모리 특정 지점에 변수를 생성하기 위해서는 **전역변수**로 선언



PUSH R28 PUSH R29

> MOVW R26, R24 MOVW R30, R22

- 레지스터 이름 설정
- 전체 암호와 알고리즘 구조
 - 빨강: 스택 관리
 - 파랑: 로드, 스토어
 - 초록: 라운드 함수

#define X00 R18 #define X01 R19 #define X10 R20 #define X11 R21 #define X20 R22 #define X21 R23 #define X30 R24 #define X31 R25

#define TM0 R26 #define TM1 R27

#define RC R29 #define RK R0

#define CNT R28

LD X00, X+ LD X01, X+ LD X10, X+ LD X11, X+ LD X20, X+ LD X21, X+ LD X30, X+ LD X31, X+

PUSH R26 PUSH R27

CLR RC LDI CNT, 88

LOOP: ANDI R30, 31 ODD_ROUND EVEN_ROUND

CPSE RC, CNT RJMP LOOP

POP R27 POP R26

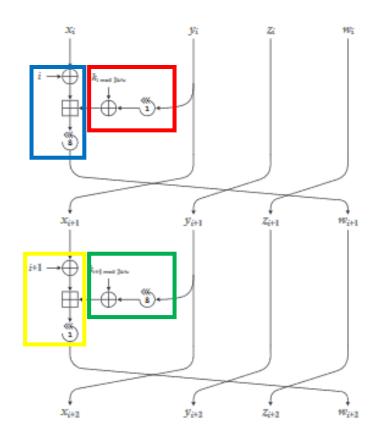
ST -X, X31 ST -X, X30 ST -X, X21 ST -X, X20 ST -X, X11 ST -X, X10 ST -X, X01 ST -X, X00

POP R29 POP R28

RET



• 홀수 라운드와 짝수 라운드



```
.macro ODD_ROUND
   MOVW TM0, X10
   LSL TM0
   ROL TM1
   ADC TM0, R1
   LD RK, Z+
   EOR TM0, RK
   LD RK, Z+
   EOR TM1, RK
   EOR X00, RC
   ADD X00, TM0
   ADC X01, TM1
   MOV TM0, X00
   MOV X00, X01
   MOV X01, TM0
   MOVW TM0, X00
   MOVW X00, X10
   MOVW X10, X20
   MOVW X20, X30
   MOVW X30, TM0
   INC RC
```

.endm

```
.macro EVEN_ROUND
   MOV TM0, X11
   MOV TM1, X10
   LD RK, Z+
   EOR TM0, RK
   LD RK, Z+
   EOR TM1, RK
   EOR X00, RC
   ADD X00, TM0
   ADC X01, TM1
   LSL X00
   ROL X01
   ADC X00, R1
   MOVW TM0, X00
   MOVW X00, X10
   MOVW X10, X20
   MOVW X20, X30
   MOVW X30, TM0
   INC RC
.endm
```

Q&A

