.text0000850C ; ---------------------------------------------------------------------------

.text0000850C

.text0000850C loc\_850C ; DATA XREF .got0000AFA8o

.text0000850C STMFD SP!, {R4-R8,LR}

.text00008510 LDR R3, =(aHelloworld - 0x8524) ; helloworld字符串的相对地址

.text00008514 LDR LR, =(unk\_9B38 - 0x854C) ; 一组特殊数据的相对地址

.text00008518 SUB SP, SP, #0x48

.text0000851C ADD R3, PC, R3 ; 相对地址+当前指令地址=绝对地址，现在可以通过r3访问到字符串了

.text00008520 MOV R6, R1

.text00008524 ADD R5, SP, #0x2C

.text00008528 LDMIA R3, {R0-R2} ; 读取r3寄存器中的数据到r0,r1,r2中

.text0000852C MOV R3, R2,LSR#16 ; 将r2的中值逻辑右移16位，再赋值给r3

.text00008530 STR R0, [SP,#0x2C] ; 将寄存器中的值存入到内存中

.text00008534 STR R1, [R5,#4] ; R5=SP+0X2C,R5+4刚好在上一个指令SP+0X2C之后

.text00008538 STRH R2, [SP,#0x34] ; 将R2的低位字节存入内存中，H表示无符号半字

.text0000853C STRB R3, [SP,#0x36] ; 将R3的值按无符号字节放入内存，B代表无符号字节

.text00008540 ADD R12, SP, #4 ; 栈空间向上移动4，地址存入R12，便于后面存放数据

.text00008544 ADD LR, PC, LR ; 特殊数据的相对地址+当前地址=特殊数据的绝对地址，现在可以访问其中的数据 了

.text00008548 LDMIA LR!, {R0-R3} ; 读取前四个数据的值到R0-R3四个寄存器中

.text0000854C STMIA R12!, {R0-R3} ; 将读取到的数值放入内存，以便空出寄存器，继续读取数据

.text00008550 LDMIA LR!, {R0-R3} ; 继续读取数据

.text00008554 LDR R4, =(off\_AFAC - 0x8564)

.text00008558 STMIA R12!, {R0-R3} ; 读入数据到内存的同时，R12寄存器加上4寄存器个数的偏移

.text0000855C LDR R4, [PC,R4]

.text00008560 LDR R0, [LR] ; 读取特殊字符串中的倒数第二个数

.text00008564 LDR R1, [LR,#4] ; 读取特殊字符串中的最后一个数据

.text00008568 LDR LR, [R4] ; 读取R4所以地址中的值到LR寄存器

.text0000856C LDR R2, =0x2AAAAAAB ; 赋值一个常量

.text00008570 ADD R3, SP, #4

.text00008574 STR LR, [SP,#0x44]

.text00008578 STR R0, [R12] ; 存储倒数第二个数据到内存

.text0000857C STR R1, [R12,#4] ; 存储最后一个数据到内存

.text00008580 ADD R0, SP, #0x2B ; 获取字符串的首地址，并存入到R0寄存器中

.text00008584 MOV LR, R3 ; 获取特殊数据串的首地址，并放入LR寄存器中

.text00008588

.text00008588 loc\_8588 ; CODE XREF .text000085ACj

.text00008588 LDR R1, [R3] ; 取R3所指地址中的数据，放入R0寄存器中，如第一个数为0xB

.text0000858C LDRB R12, [R0,#1]! ; 取字符串中的一个字符，放入到R12寄存器中

.text00008590 EOR R1, R1, #7 ; 将特殊数据串中取得的数据与7进行异或运算。

.text00008594 SMULL R8, R7, R2, R1 ; 除法变乘法，相当于R16,R7存放结果

.text00008598 SUB R1, R7, R1,ASR#31 ; 算法优化，原文是+22,再-24，这里处理为直接-2,表达式等价于R1=R7-R131

.text0000859C SUB R1, R1, #2 ; 减2

.text000085A0 EOR R1, R1, R12 ; R1与字符串中的一个字符进行异或运算

.text000085A4 STR R1, [R3],#4 ; 将得到的数值存入R3所在的地址，完成后R3的值加4

.text000085A8 CMP R3, R5 ; 比较R3与R5的值，R3是特殊数据字串的起始位置，R5是结束位置,如果两者相等，证明数据取完了

.text000085AC BNE loc\_8588 ; 取R3所指地址中的数据，放入R0寄存器中，如第一个数为0xB

.text000085B0 MOV R3, SP ; 计算出的数据的超始位置-4赋值给R3

.text000085B4 ADD R2, SP, #0x37 ; 找到一块空地，便于下面赋值

.text000085B8 ADD LR, LR, #0x24 ; 将计算出的字符串的末地址赋给LR，作为后面赋值的结束点

.text000085BC

.text000085BC loc\_85BC ; CODE XREF .text000085C8j

.text000085BC LDR R1, [R3,#4]! ; 将计算出的一个数据放入R1中

.text000085C0 CMP R3, LR ; 比较开始和结束的位置，以便判断是否赋值结束

.text000085C4 STRB R1, [R2,#1]! ; 将结果放入上面找到的空地处，也就是新字符串变量中

.text000085C8 BNE loc\_85BC ; 将计算出的一个数据放入R1中

.text000085CC MOV R3, #0 ; 末尾置0，以便让字符串中不出现垃圾字符

.text000085D0 ADD R0, SP, #0x38

.text000085D4 LDR R1, [R6,#4]

.text000085D8 STRB R3, [SP,#0x42]

.text000085DC BL \_83F0 ; 调用string.h中的函数strcmp比较两个字符串的值

.text000085E0 CMP R0, #0 ; 如果字符串相同，则注册成功

.text000085E4 BEQ loc\_8610 ; 取成功提示字符串的相对地址

.text000085E8 LDR R0, =(aNoYouAreFailed - 0x85F4) ; 取失败提示字符串的相对地址

.text000085EC ADD R0, PC, R0 ; NO,you are failed!

.text000085F0 BL \_83F\_0

.text000085F4

.text000085F4 loc\_85F4 ; CODE XREF .text0000861Cj

.text000085F4 LDR R2, [SP,#0x44]

.text000085F8 LDR R3, [R4]

.text000085FC MOV R0, #0

.text00008600 CMP R2, R3

.text00008604 BNE loc\_8620

.text00008608 ADD SP, SP, #0x48

.text0000860C LDMFD SP!, {R4-R8,PC}

.text00008610 ; ---------------------------------------------------------------------------

.text00008610

.text00008610 loc\_8610 ; CODE XREF .text000085E4j

.text00008610 LDR R0, =(aGoodWorkYouHav - 0x861C) ; 取成功提示字符串的相对地址

.text00008614 ADD R0, PC, R0 ; Good Work,you have Successed!

.text00008618 BL \_83F\_0

.text0000861C B loc\_85F4

#include stdio.h

#include string.h

int main(int argc,char argv[])

{

char name[]=helloworld;

int keys[]={0xb,0x1f,0x19,0x19,0x49,0xb,0xb,0xb,0x31,0x53};

char Thekeys[11];

int i;

for(i=0;i10;i++)

{

keys[i]^=7;

keys[i]=keys[i]6;

keys[i]+=22;

keys[i]-=24;

keys[i]^=name[i];

}

for(i=0;i10;i++)

{

Thekeys[i]=keys[i];

}

Thekeys[i]=0;

if(!strcmp(Thekeys,argv[1]))

printf(Good Work,you have Successed!);

else

printf(NO,you are failed!);

return 0;

}