سیدمحمدرضا حسینی - 97243129
سیدعباس میرقاسمی - 97243068

در این تمرین یک سیستم کامل برای اجرای هم طراحی رمزنگاری به شیوه RSA را پیاده سازی کردیم.
با توجه به اینکه در صورت تمرین اعلام شده بود باید از شیوه memory-mapped برای برقراری ارتباط میان سخت افزار و نرم افزار استفاده کنیم ما از همین شیوه memory-mapped استفاده کردیم و مانند مثال کتاب آدرس هایی از حافظه را انتخاب کردیم تا میان سخت افزار و نرم افزار بتوانیم ارتباطی ایجاد کنیم.
همچنین طبق مثال کتاب تعریف های حافظه نیز در کد gezel انجام شد. همچنین متغیرهای volatile نیز در سمت نرم‌افزار تعریف کردیم که با استفاده از اشاره گر به خانه های حافظه متناظر با آن اشاره می کرد.
حال فقط نیاز داشتیم تا یک پروتکل ارتباطی بین سخت افزار و نرم افزار را طراحی و پیاده سازی کنیم. در این حالت ما باید به سخت افزار به هنگام آماده شدن دیتا در نرم‌افزار اطلاع بدهیم تا پارامترهای ورودی بگیرد به همین علت نیز پورتی به نام req قرار دادیم تا این آماده شدن دیتا را به سخت افزار اطلاع بدهد. حال بعد از خواندن پارامترهای ورودی از حافظه نیاز بود تا ورودی داده شده را رمزگذاری کنیم و بعد از آن نیز آن را رمز گشایی کنیم تا زمانی که خروجی های ما شامل cipher و plaint که حاصل از رمزگذاری و رمزگشایی هستند آماده شوند لازم است تا بخش نرم‌افزاری صبر کند. برای همین یک پورت دیگر نیز به نام ack میان سخت افزار و نرم افزار تعریف کردیم تا بر اساس وضعیتی که ماشین حالت سخت افزار تعیین می‌کند نرم افزار صبر کند تا نتایج آماده شود و در خانه های حافظه قرار بگیرد. همچنین برای سادگی تمام ورودی هایی که لازم بود از طریق نرم افزار به سخت افزار داده شود را نیز به صورت جدا در حافظه قرارداد دادیم. برای اجرای ما دو بخش مجزا برای اجرا داشتیم. یک بخش کد c و تبدیل آن به یک برنامه برای کد fdl. و بخش دیگر اجرای کد fdl و خروجی گرفتن از آن.

از دستورهای زیر برای این کار استفاده کردیم:

```
arm-linux-gcc -static main.c -o main
gplatform model.fdl
```

همچنین برای اجرای این برنامه نیاز به نصب برنامه هایی در سیستم عامل بود که ما آن را انجام دادیم و در نهایت یک ایمیج داکری برای ساده تر شدن وضعیتمان ایجاد کردیم تا راحت تر بتوانیم کد های مان را کامپایل و اجرا کنیم:

ایمیج داکری ایجاد شده:

```
samirghasemi/gezel:3.1.1
docker run -it -v $HOME/Documents/gezel:/opt/src samirghasemi/gezel:3.1.1
```

نکته:

کد fdl و کد c به صورت کامل کامنت گذاری شده است. این فایل ها نیز به پیوست خدمت تان ارسال میگردد.

کد c

commands.sh | model.fdl M | main.c

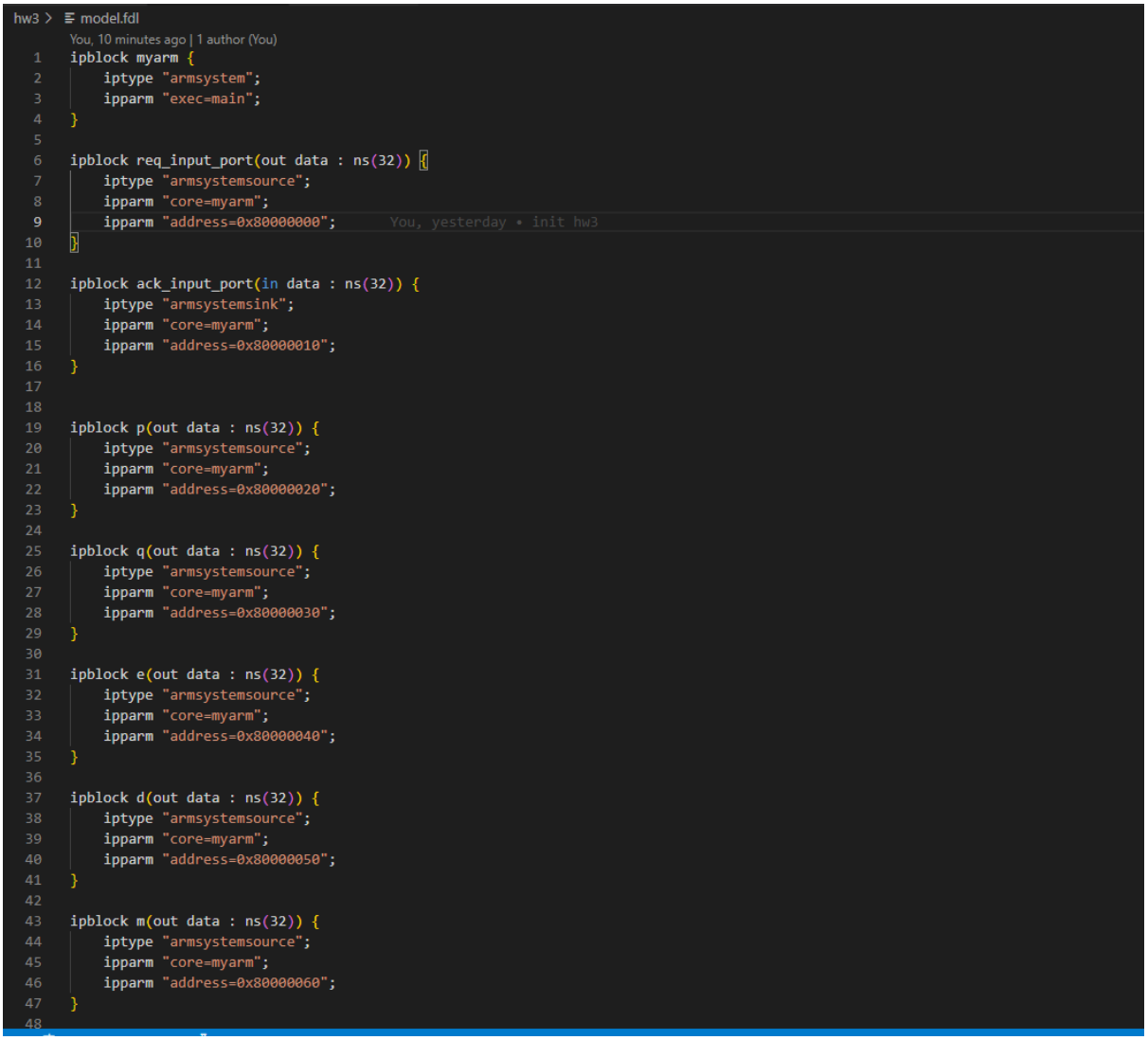
hw3 > main.c > encrypted_data

You, 21 hours ago | 1 author (You)

```c
#include <stdio.h>

volatile unsigned int *req =            (unsigned int *) 0x80000000;
volatile unsigned int *ack =            (unsigned int *) 0x80000010;

volatile unsigned int *p =              (unsigned int *) 0x80000020;
volatile unsigned int *q =              (unsigned int *) 0x80000030;
volatile unsigned int *e =              (unsigned int *) 0x80000040;
volatile unsigned int *d =              (unsigned int *) 0x80000050;
volatile unsigned int *m =              (unsigned int *) 0x80000060;

volatile unsigned int *encrypted_data = (unsigned int *) 0x80000070;        You, 21 hours ago •
volatile unsigned int *decrypted_data = (unsigned int *) 0x80000080;

void req1() {
    *req = 1;
}

void await1(){
    while (*ack == 1);
}

void await2(){
    while (*ack == 2);
}

int main(){
    *p = 23;
    *q = 2;
    *e = 17;
    *d = 31;
    *m = 20;

    req1();
    await1();
    // doing encryption and decryption
    await2();

    // print results of encryption and decryption
    printf("SOFTWARE----> encrypted data = %d \n", *encrypted_data);
    printf("SOFTWARE----> decrypted data = %d \n", *decrypted_data);

}
```

کد fdl:

بخشی از پیاده سازی ipblock برای اختصاص حافظه مشترک بین سخت افزار و نرم افزار




اتصالات اولیه:

```
dp RSA_encryptor_decryptor {
    use myarm;

    // initialize signals
    sig req_sig , ack_sig : ns(32);
    sig encrypted_sig , decrypted_sig : ns(32);
    sig p_sig, q_sig, e_sig, d_sig, m_sig : ns(32);
    // connect signals to inputs
    use req_input_port(req_sig);
    use ack_input_port(ack_sig);
    use p(p_sig);
    use q(q_sig);
    use e(e_sig);
    use d(d_sig);
    use m(m_sig);
    use encrypted_data_output_port(encrypted_sig);
    use decrypted_data_output_port(decrypted_sig);

    // initialize registers
    reg encrypted_reg , decrypted_reg : ns(32);
    reg current_request_reg , done_reg , modulo_reg : ns(1);

    reg x, y, p, q, e, d, n, z : ns(32);
    reg message , cipher , plaint : ns(32);

    always{
        current_request_reg = req_sig;
        encrypted_sig = encrypted_reg;
        decrypted_sig = decrypted_reg;
    }
```

Sfg برای مقداردهی ack ها و خواندن دیتا اولیه از روی پورت ها:

```
    //// set acknowledges signals

    sfg ack1 {
        ack_sig = 1;
    }

    sfg ack2 {
        ack_sig = 2;      You, yesterday • init hw3 …
    }

    sfg ack3 {
        ack_sig = 3;
    }

    //// read data from inputs

    sfg data_reader {
        p = p_sig;
        q = q_sig;
        e = e_sig;
        d = d_sig;
        message = m_sig;
        done_reg = 0;
        $display("HARDWARE----> get data from software ", " p:", p, " q:", q, " e:", e, " d:", d, " msg:", message);
    }
```

Sfg برای انجام محاسبات اولیه در رمزگذاری و رمزگشایی

```
    //// calculations      You, 21 hours ago • update …
    sfg calculate_n_and_z {
        n = p * q;
        z = ( p - 1 ) * ( q - 1 );

        $display("HARDWARE----> calculate n and z", " n:", n, " z:", z);
        $display($dec);
    }
    sfg initialize_for_start {
        x = e;
        y = z;
    }
    sfg calculate_gcd {
        done_reg = ( (x == 0) | (y == 0) );
        encrypted_reg = x > y ? x : y;
        x = x > y ? x - y : x;
        y = y >= x ? y - x : y;

        $display("HARDWARE----> calculate gcd", " x:", x, " y:", y);
    }
    sfg check_encrypted_reg {
        done_reg = encrypted_reg == 1 ? 1 : 0 ;
    }
    sfg increment1 {
        x = e + 1;
        y = z;
        e = e + 1;
    }
    sfg initialize_for_after_start {
        x = d * e;
        y = z;
    }
    sfg calculate_x {
        done_reg = x < y ? 1 : 0;
        x = x >= y ? x - y : x;

        $display("HARDWARE----> calculate x", " x:", x);
    }
    sfg check_done_reg {
        done_reg = x == 1 ? 1 : 0 ;
    }
    sfg increment2 {
        x = ( d + 1 ) * e;
        y = z;
        d = d + 1;
```

Sfg های بخش رمزگذاری:

```
//// encryption

sfg initialize_for_encryption {
    x = 1;
    y = e - 1;
}

sfg encrypter {
    done_reg = y == 0 ? 1 : 0;
    x = x * message;
    y = y - 1;

    $display("HARDWARE----> encryption", " x:", x, " y:", y);
}

sfg loop_for_module {
    x = x >= n ? x - n : x;
    modulo_reg = x > n ? 0 : 1;
}

sfg set_cipher_reg {
    cipher = x;
}

sfg set_encrypted_reg {
    encrypted_reg = cipher;
}
```

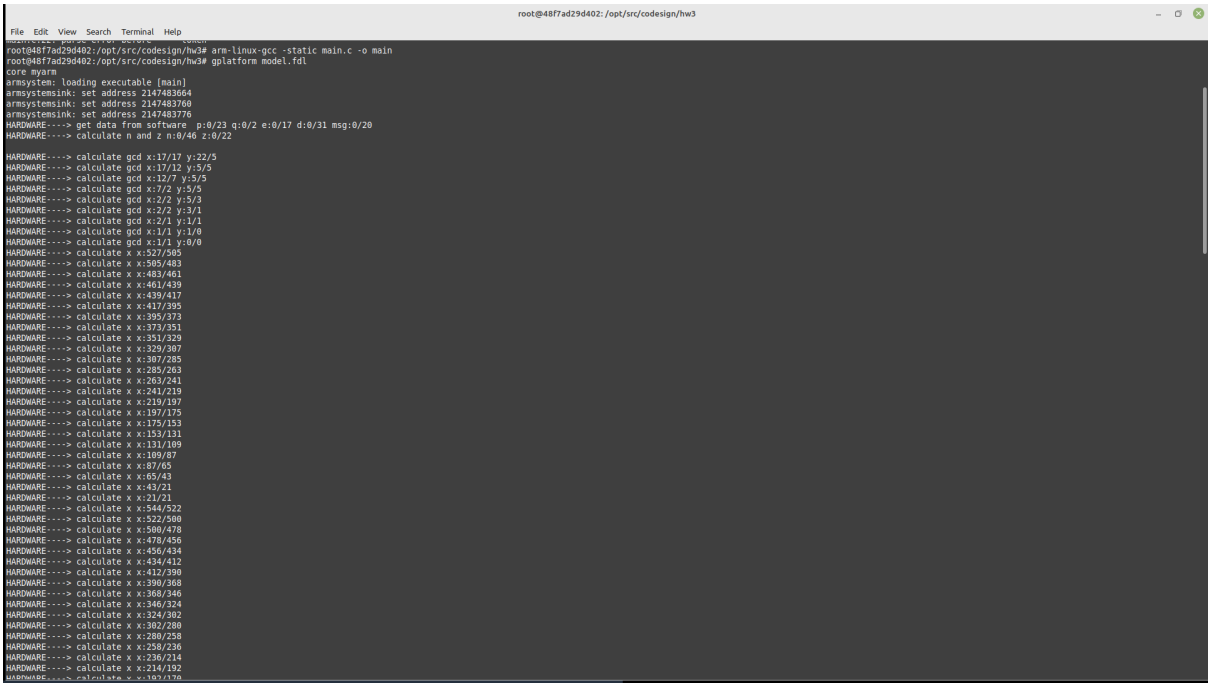
Sfg های بخش رمزگشایی

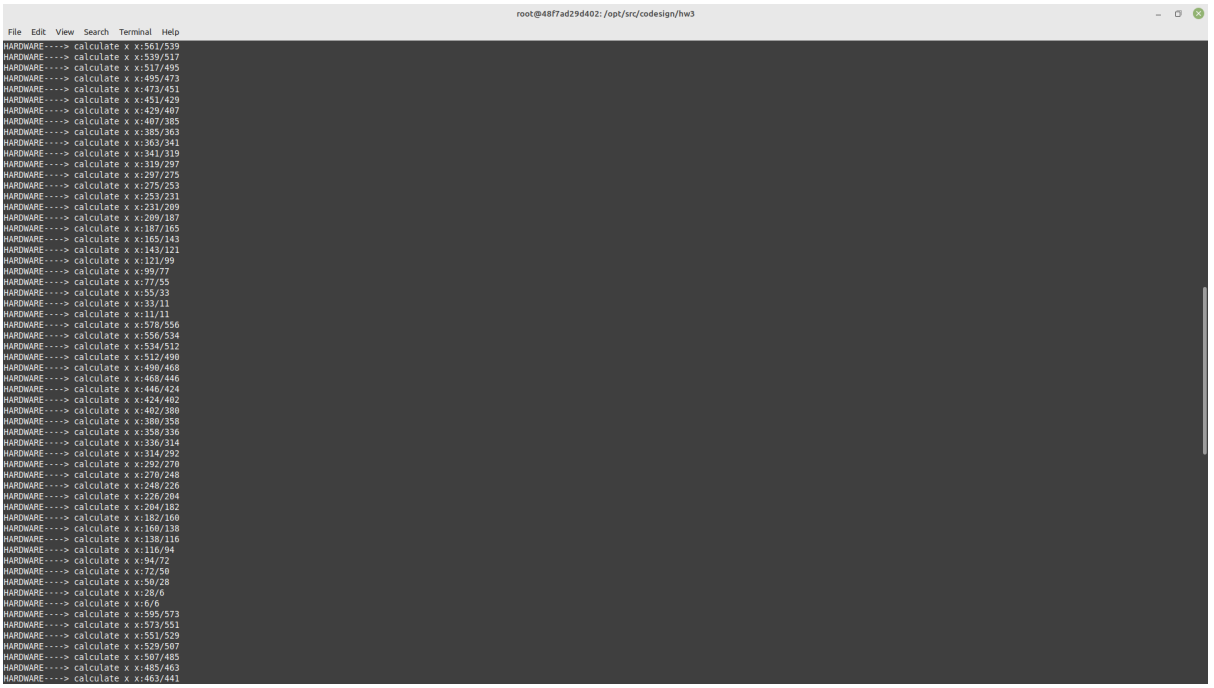
```
//// decryption

sfg initialize_for_decryption {
    x = 1;
    y = d-1;
    modulo_reg = 0;
    done_reg = 0;
}

sfg decrypter {
    done_reg = y == 0 ? 1 : 0;
    x = x * cipher;
    y = y - 1;

    $display("HARDWARE----> doing_decryption", " x:", x, " y:", y);
}

sfg set_plaint_reg {
    plaint = x;
}

sfg set_decrypted_reg{
    decrypted_reg = plaint;
}
```

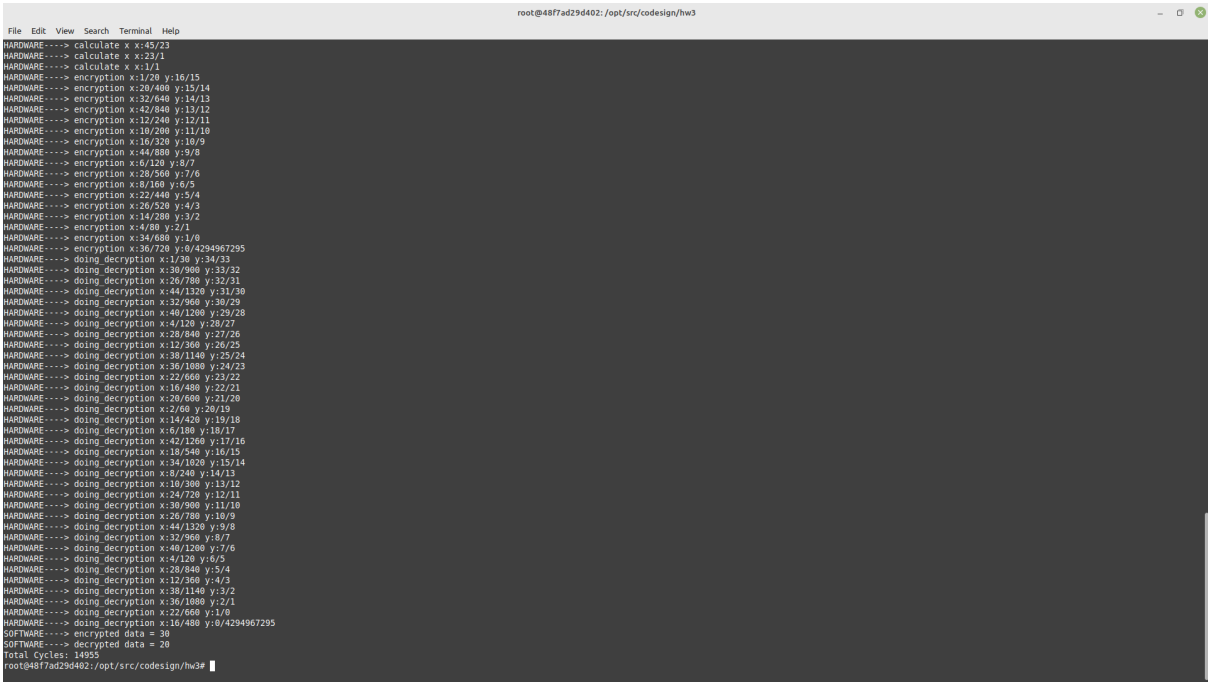
بخش state ها:

```
fsm rsaController(RSA_encryptor_decryptor){
    initial s1;
    state   s2, s3, s4, s5, s6, s7, s8, s9, s10, s11, s12, s13, s14, s15, s16, s17, s18, s19, s20, s21, s22, s23, s24;
    @s1 if(current_request_reg) then (data_reader, ack2) -> s2;
    else (ack1) ->s1;
    @s2 (calculate_n_and_z, ack2) -> s3;
    @s3 (initialize_for_start, ack2) -> s4;
    @s4 (calculate_gcd, ack2) -> s5;
    @s5 if(done_reg) then (check_encrypted_reg, ack2) -> s6;
    else (calculate_gcd, ack2) -> s5;
    @s6 if(~done_reg) then (increment1, ack2) -> s4;
    else (ack2) -> s7;
    @s7 (initialize_for_after_start, ack2) -> s8;
    @s8 (calculate_x, ack2) -> s9;
    @s9 if(done_reg) then (check_done_reg, ack2) -> s10;
    else (calculate_x, ack2) -> s9;
    @s10 if(~done_reg) then (increment2, ack2) -> s8;
    else (ack2) -> s11;
    @s11 (ack2) -> s12;
    @s12 (initialize_for_encryption, ack2) -> s13;
    @s13 (encrypter, ack2) -> s14;
    @s14 (loop_for_module, ack2) -> s15;
    @s15 if(modulo_reg) then (ack2) -> s16;
    else (ack2) -> s14;
    @s16 if(done_reg) then (set_cipher_reg, ack2) -> s17;
    else (ack2) -> s13;
    @s17 (set_encrypted_reg, ack2) -> s18;
    @s18 (initialize_for_decryption, ack2) -> s19;
    @s19 (decrypter, ack2) -> s20;
    @s20 (loop_for_module, ack2) -> s21;
    @s21 if(modulo_reg) then (ack2) -> s22;
    else (ack2) -> s20;
    @s22  if(done_reg) then (set_plaint_reg, ack2) -> s23;
    else (ack2) -> s19;
    @s23 (ack3, set_decrypted_reg) -> s24;
    @s24 (ack3) -> s24;
}

system S {
    RSA_encryptor_decryptor;
}

```

خروجی های شبیه سازی شده به شرح زیر می باشند:


root@48f7ad29d402: /opt/src/codesign/hw3
File Edit View Search Terminal Help
root@48f7ad29d402:/opt/src/codesign/hw3# arm-linux-gcc -static main.c -o main
root@48f7ad29d402:/opt/src/codesign/hw3# gplatform model.fdl
core myarm
armsystem: loading executable [main]
armsystemsink: set address 2147483664
armsystemsink: set address 2147483760
armsystemsink: set address 2147483776
HARDWARE----> get data from software  p:0/23 q:0/2 e:0/17 d:0/31 msg:0/20
HARDWARE----> calculate n and z n:0/46 z:0/22
HARDWARE----> calculate gcd x:17/17 y:22/5
HARDWARE----> calculate gcd x:17/12 y:5/5
HARDWARE----> calculate gcd x:12/7 y:5/5
HARDWARE----> calculate gcd x:7/2 y:5/5
HARDWARE----> calculate gcd x:2/2 y:5/3
HARDWARE----> calculate gcd x:2/2 y:3/1
HARDWARE----> calculate gcd x:2/1 y:1/1
HARDWARE----> calculate gcd x:1/1 y:1/0
HARDWARE----> calculate gcd x:1/1 y:0/0
HARDWARE----> calculate x x:527/505
HARDWARE----> calculate x x:505/483
HARDWARE----> calculate x x:483/461
HARDWARE----> calculate x x:461/439
HARDWARE----> calculate x x:439/417
HARDWARE----> calculate x x:417/395
HARDWARE----> calculate x x:395/373
HARDWARE----> calculate x x:373/351
HARDWARE----> calculate x x:351/329
HARDWARE----> calculate x x:329/307
HARDWARE----> calculate x x:307/285
HARDWARE----> calculate x x:285/263
HARDWARE----> calculate x x:263/241
HARDWARE----> calculate x x:241/219
HARDWARE----> calculate x x:219/197
HARDWARE----> calculate x x:197/175
HARDWARE----> calculate x x:175/153
HARDWARE----> calculate x x:153/131
HARDWARE----> calculate x x:131/109
HARDWARE----> calculate x x:109/87
HARDWARE----> calculate x x:87/65
HARDWARE----> calculate x x:65/43
HARDWARE----> calculate x x:43/21
HARDWARE----> calculate x x:21/21
HARDWARE----> calculate x x:544/522
HARDWARE----> calculate x x:522/500
HARDWARE----> calculate x x:500/478
HARDWARE----> calculate x x:478/456
HARDWARE----> calculate x x:456/434
HARDWARE----> calculate x x:434/412
HARDWARE----> calculate x x:412/390
HARDWARE----> calculate x x:390/368
HARDWARE----> calculate x x:368/346
HARDWARE----> calculate x x:346/324
HARDWARE----> calculate x x:324/302
HARDWARE----> calculate x x:302/280
HARDWARE----> calculate x x:280/258
HARDWARE----> calculate x x:258/236
HARDWARE----> calculate x x:236/214
HARDWARE----> calculate x x:214/192


root@48f7ad29d402: /opt/src/codesign/hw3
File Edit View Search Terminal Help
HARDWARE----> calculate x x:561/539
HARDWARE----> calculate x x:539/517
HARDWARE----> calculate x x:517/495
HARDWARE----> calculate x x:495/473
HARDWARE----> calculate x x:473/451
HARDWARE----> calculate x x:451/429
HARDWARE----> calculate x x:429/407
HARDWARE----> calculate x x:407/385
HARDWARE----> calculate x x:385/363
HARDWARE----> calculate x x:363/341
HARDWARE----> calculate x x:341/319
HARDWARE----> calculate x x:319/297
HARDWARE----> calculate x x:297/275
HARDWARE----> calculate x x:275/253
HARDWARE----> calculate x x:253/231
HARDWARE----> calculate x x:231/209
HARDWARE----> calculate x x:209/187
HARDWARE----> calculate x x:187/165
HARDWARE----> calculate x x:165/143
HARDWARE----> calculate x x:143/121
HARDWARE----> calculate x x:121/99
HARDWARE----> calculate x x:99/77
HARDWARE----> calculate x x:77/55
HARDWARE----> calculate x x:55/33
HARDWARE----> calculate x x:33/11
HARDWARE----> calculate x x:11/11
HARDWARE----> calculate x x:578/556
HARDWARE----> calculate x x:556/534
HARDWARE----> calculate x x:534/512
HARDWARE----> calculate x x:512/490
HARDWARE----> calculate x x:490/468
HARDWARE----> calculate x x:468/446
HARDWARE----> calculate x x:446/424
HARDWARE----> calculate x x:424/402
HARDWARE----> calculate x x:402/380
HARDWARE----> calculate x x:380/358
HARDWARE----> calculate x x:358/336
HARDWARE----> calculate x x:336/314
HARDWARE----> calculate x x:314/292
HARDWARE----> calculate x x:292/270
HARDWARE----> calculate x x:270/248
HARDWARE----> calculate x x:248/226
HARDWARE----> calculate x x:226/204
HARDWARE----> calculate x x:204/182
HARDWARE----> calculate x x:182/160
HARDWARE----> calculate x x:160/138
HARDWARE----> calculate x x:138/116
HARDWARE----> calculate x x:116/94
HARDWARE----> calculate x x:94/72
HARDWARE----> calculate x x:72/50
HARDWARE----> calculate x x:50/28
HARDWARE----> calculate x x:28/6
HARDWARE----> calculate x x:6/6
HARDWARE----> calculate x x:595/573
HARDWARE----> calculate x x:573/551
HARDWARE----> calculate x x:551/529
HARDWARE----> calculate x x:529/507
HARDWARE----> calculate x x:507/485
HARDWARE----> calculate x x:485/463
HARDWARE----> calculate x x:463/441


root@48f7ad29d402: /opt/src/codesign/hw3
File Edit View Search Terminal Help
HARDWARE----> calculate x x:45/23
HARDWARE----> calculate x x:23/1
HARDWARE----> calculate x x:1/1
HARDWARE----> encryption x:1/20 y:16/15
HARDWARE----> encryption x:20/400 y:15/14
HARDWARE----> encryption x:32/640 y:14/13
HARDWARE----> encryption x:42/840 y:13/12
HARDWARE----> encryption x:12/240 y:12/11
HARDWARE----> encryption x:10/200 y:11/10
HARDWARE----> encryption x:16/320 y:10/9
HARDWARE----> encryption x:44/880 y:9/8
HARDWARE----> encryption x:6/120 y:8/7
HARDWARE----> encryption x:28/560 y:7/6
HARDWARE----> encryption x:8/160 y:6/5
HARDWARE----> encryption x:22/440 y:5/4
HARDWARE----> encryption x:26/520 y:4/3
HARDWARE----> encryption x:14/280 y:3/2
HARDWARE----> encryption x:4/80 y:2/1
HARDWARE----> encryption x:34/680 y:1/0
HARDWARE----> encryption x:36/720 y:0/4294967295
HARDWARE----> doing_decryption x:1/30 y:34/33
HARDWARE----> doing_decryption x:30/900 y:33/32
HARDWARE----> doing_decryption x:26/780 y:32/31
HARDWARE----> doing_decryption x:44/1320 y:31/30
HARDWARE----> doing_decryption x:32/960 y:30/29
HARDWARE----> doing_decryption x:40/1200 y:29/28
HARDWARE----> doing_decryption x:4/120 y:28/27
HARDWARE----> doing_decryption x:28/840 y:27/26
HARDWARE----> doing_decryption x:12/360 y:26/25
HARDWARE----> doing_decryption x:38/1140 y:25/24
HARDWARE----> doing_decryption x:36/1080 y:24/23
HARDWARE----> doing_decryption x:22/660 y:23/22
HARDWARE----> doing_decryption x:16/480 y:22/21
HARDWARE----> doing_decryption x:20/600 y:21/20
HARDWARE----> doing_decryption x:2/60 y:20/19
HARDWARE----> doing_decryption x:14/420 y:19/18
HARDWARE----> doing_decryption x:6/180 y:18/17
HARDWARE----> doing_decryption x:42/1260 y:17/16
HARDWARE----> doing_decryption x:18/540 y:16/15
HARDWARE----> doing_decryption x:34/1020 y:15/14
HARDWARE----> doing_decryption x:8/240 y:14/13
HARDWARE----> doing_decryption x:10/300 y:13/12
HARDWARE----> doing_decryption x:24/720 y:12/11
HARDWARE----> doing_decryption x:30/900 y:11/10
HARDWARE----> doing_decryption x:26/780 y:10/9
HARDWARE----> doing_decryption x:44/1320 y:9/8
HARDWARE----> doing_decryption x:32/960 y:8/7
HARDWARE----> doing_decryption x:40/1200 y:7/6
HARDWARE----> doing_decryption x:4/120 y:6/5
HARDWARE----> doing_decryption x:28/840 y:5/4
HARDWARE----> doing_decryption x:12/360 y:4/3
HARDWARE----> doing_decryption x:38/1140 y:3/2
HARDWARE----> doing_decryption x:36/1080 y:2/1
HARDWARE----> doing_decryption x:22/660 y:1/0
HARDWARE----> doing_decryption x:16/480 y:0/4294967295
SOFTWARE----> encrypted data = 30
SOFTWARE----> decrypted data = 20
Total Cycles: 14955
root@48f7ad29d402:/opt/src/codesign/hw3#