باسمه تعالی

هم‌طراحی سخت‌افزار و نرم‌افزار

(سال تحصیلی ۱۴۰۲-۱۴۰۱، نیم‌سال دوم)

تمرین چهارم: Implementing a System Using Hardware/Software Codesign

هدف از انجام این تمرین شبیه‌سازی یک سیستم متشکل از سخت‌افزار و نرم‌افزار با استفاده از محیط شبیه‌سازی GEZEL، می باشد. به این منظور قصد داریم، کاربرد رمزگذاری و رمزگشایی به روش RSA را به صورت هم‌طراحی سخت افزار و نرم افزار و با استفاده از پردازنده ARM، شبیه‌سازی نماییم.

RSA شیوه‌ای برای رمزنگاری به روش کلید عمومی است. رمزگذاری نامتقارن یا کلید عمومی بر اساس دو کلید به نام‌های عمومی و خصوصی صورت می‌گیرد. کلید، عددی ثابت است که در محاسبات رمزنگاری استفاده می‌شود. کلید عمومی برای همه معلوم بوده و برای رمزنگاری پیام استفاده می‌شود. این پیام فقط توسط کلید خصوصی باز می‌شود. به بیان دیگر همه می‌توانند یک پیام را رمز کنند اما فقط صاحب کلید خصوصی می‌تواند پیام را باز کند و بخواند. هر چند از لحاظ ریاضی کلیدهای عمومی و خصوصی با یکدیگر ارتباط دارند اما تقریباً محال است که کسی بتواند حتی با تجهیزات پیشرفته و صرف وقت زیاد با داشتن یکی از کلیدها، دیگری را تشخیص دهد .

الگوریتم RSA در نهایت سادگی به صورت زیر است:

۱. پیامی که باید رمز شود به بلوکهای K کاراکتری تقسیم بندی می شود.

۲. هر بلوک طبق قاعده ای کاملا دلخواه به یک عدد صحیح به نام P تبدیل می گردد.

۳. با جفت عدد (e,n) به ازای یکایک بلوکهای P اعداد جدیدی طبق رابطه زیر بدست می آیند:

$$c = m^e \mod n$$

۴. کدهای C بجای کدهای اصلی P ارسال می شوند.

در روش رمزگشایی داده ها نیز با داشتن جفت عدد (d,n) بلوکهای رمز شده بصورت زیر از رمز خارج می شوند:

$$m = c^d \mod n$$

روش انتخاب e و dکه توسط ابداع کنندگان RSA پیشنهاد شده ،عبارت است از:

۱. دو عدد دلخواه (اما بزرگ) p و q انتخاب می شوند.

۲. اعداد n و z طبق دو رابطه زیر محاسبه می گردند:

$$n = pq$$
$$z = (p-1)(q-1)$$

۳. عدد d طوری انتخاب می شود که نسبت به z اول و از آن کوچکتر باشد .

۴. بر اساس d ،عدد e طوری انتخاب می شود که رابطه زیر برقرار باشد:

$$de \; mod \; z = 1$$

در ادامه یک نمونه پیاده‌سازی از الگوریتم RSA به زبان C را مشاهده می‌نماییم. در این کد ۳ مقدار برای p، q و m در نظر گرفته شده است. **(p , q باید اول باشند)**

```c
#include<stdio.h>
// Returns gcd of a and b
unsigned int gcd(unsigned int a, unsigned int h)
{
    unsigned int temp;
    while (1)
    {
        temp = a%h;
        if (temp == 0)
          return h;
        a = h;
        h = temp;
    }
}
unsigned long long power(unsigned long long a, unsigned int n)
{
    unsigned int i;
    unsigned long long pow=a;
    for (i=1; i<n; i++)
    {
        pow = pow * a;
    }

    return pow;
}
```

```c
int main()
{
    // Two random prime numbers
    unsigned int p = 23;
    unsigned int q = 2;
    // First part of public key:
    unsigned int n = p*q;
    // Finding other part of public key.
    unsigned int e = 2;
    unsigned int z = (p-1)*(q-1);

    // point1

    while (e < z)
    {
        // e must be co-prime to z and smaller than z.
        if (gcd(e, z)==1)
            break;
        else
            e++;
    }

    // point2

    // choosing d such that it satisfies d*e = 1 + k * totient
    unsigned int d = 2;
    while (d < z)
    {
        if ((d*e)%z == 1)
            break;
        else
            d++;
    }

    // point3

    // Message to be encrypted
    unsigned long long msg = 26;
    printf("Message data = %ld", msg);

    // point4

    // Encryption c = (msg ^ e) % n
    unsigned long long c = power(msg, e);
    c = c%n;
    printf("\nEncrypted data = %ld", c);

    // point5

    // Decryption m = (c ^ d) % n
    unsigned long long m = power(c, d);
    m = m%n;
    printf("\nOriginal Message Sent = %ld", m);

    return 0;
}
```

۱. در این پروژه باید الگوریتم فوق را به صورت هم‌طراحی سخت‌افزار و نرم‌افزار، در محیط شبیه‌سازی GEZEL با استفاده از پردازنده ARM، پیاده‌سازی نمایید. شبیه‌ساز GEZEL امکان استفاده از یک پردازنده ARM به صورت یک ipcore و تعریف ارتباطات مورد نیاز بین سخت‌افزار تولید شده و کد نرم‌افزاری را فراهم می‌نماید. مطابق مثال ارائه شده در بخش 13.3 از کتاب مرجع، سیستم پیاده‌سازی شده با استفاده از زبان GEZEL شامل قسمت‌های ARM Core، ARM Interfaces، Hardware Kernel و Hardware Interface می‌باشد. قسمت Hardware Kernel همان کد سخت‌افزار است و Hardware Interface ارتباط ورودی و خروجی‌های سخت‌افزار با ارتباطات تعریف شده در قسمت ARM Interfaces را مشخص می‌کند. علاوه بر این، کد نرم‌افزار یاSoftware Driver نیز بایستی به زبان C و با استفاده از روش Memory-Mapped نوشته شده و فایل کامپایل شده آن همانند مثال در قسمت ARM Core معرفی شود. سرانجام پس از آماده شدن سیستم، عملیات شبیه‌سازی با استفاده از دستور gplatform انجام می‌شود.

به این منظور باید قسمتی از کد فوق را به صورت سخت‌افزاری و قسمتی را به صورت نرم‌افزاری انجام دهید و در پایان نتیجه محاسبات را در هر دو قسمت نمایش دهید. برای انجام قسمتی از کد به صورت سخت‌افزاری و استفاده از پردازنده ARM به این شکل عمل شود که تمامی بخش‌های محاسباتی در سخت‌افزار انجام شود و کنترل شروع و پایان هر قسمت که در شکل فوق نشان داده شده است و همچنین مشخص کردن مقادیر اولیه توسط نرم‌افزار صورت پذیرد. مابقی فرض‌ها از جمله تعداد و نوع پایه‌های ارتباطی می‌تواند به دلخواه انجام شود.

۲. گزارش مختصری از فرآیند انجام پروژه به همراه تصاویر مناسب، در قالب یک گزارش با فرمت خواسته شده به انضمام کدهای نوشته شده و فایل‌های خروجی ابزار را به صورت فشرده، همراه با نام و شماره دانشجویی در سامانه درس‌افزار بارگذاری نمایید (قالب گزارش از قسمت فایل‌ها قابل دسترسی است).

توجه: در گزارش نوشته شده باید تمامی فرض‌های گرفته شده توضیح داده شوند.

موفق باشید