



هم طراحی سخت افزار و نرم افزار

(سال تحصیلی ۰۰-۰۱، نیمسال دوم)

تمرین سوم: Implementing a System Using Hardware/Software Codesign

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

RSA شیوه ای برای رمزنگاری به روش کلید عمومی است. رمزگذاری نامتقارن یا کلید عمومی بر اساس دو کلید به نام های عمومی و خصوصی صورت می گیرد. کلید، عددی ثابت است که در محاسبات رمزنگاری استفاده می شود. کلید عمومی برای همه معلوم بوده و برای رمزنگاری پیام استفاده می شود. این پیام فقط توسط کلید خصوصی باز می شود. به بیان دیگر همه می توانند یک پیام را رمز کنند اما فقط صاحب کلید خصوصی می تواند 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 کاملاً 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 d که 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 پیاده سازی از الگوریتم RSA به زبان C را مشاهده می نماییم. در این کد ۳ مقدار برای p، q و

m در نظر گرفته شده است. (p, q باید نسبت به هم اول باشند)

```

1  #include<stdio.h>
2  // Returns gcd of a and b
3  unsigned int gcd(unsigned int a, unsigned int h)
4  {
5      unsigned int temp;
6      while (1)
7      {
8          temp = a%h;
9          if (temp == 0)
10             return h;
11          a = h;
12          h = temp;
13      }
14  }
15  unsigned long long power(unsigned long long a, unsigned int n)
16  {
17      unsigned int i;
18      unsigned long long pow=a;
19      for (i=1; i<n; i++)
20      {
21          pow = pow * a;
22      }
23
24      return pow;
25  }
```

```

26 int main()
27 {
28     // Two random prime numbers
29     unsigned int p = 9;
30     unsigned int q = 14;
31     // First part of public key:
32     unsigned int n = p*q;
33     // Finding other part of public key.
34     unsigned int e = 2;
35     unsigned int z = (p-1)*(q-1);
36
37     // point1
38
39     while (e < z)
40     {
41         // e must be co-prime to z and smaller than z.
42         if (gcd(e, z) == 1)
43             break;
44         else
45             e++;
46     }
47
48     // point2
49
50     // choosing d such that it satisfies  $d \cdot e = 1 + k \cdot \text{totient}$ 
51     unsigned int d = 2;
52     while (d < z)
53     {
54         if ((d*e)%z == 1)
55             break;
56         else
57             d++;
58     }
59
60     // point3
61
62     // Message to be encrypted
63     unsigned long long msg = 12;
64     printf("Message data = %ld", msg);
65
66     // point4
67
68     // Encryption  $c = (msg^e) \% n$ 
69     unsigned long long c = power(msg, e);
70     c = c%n;
71     printf("\nEncrypted data = %ld", c);
72
73     // point5
74
75     // Decryption  $m = (c^d) \% n$ 
76     unsigned long long m = power(c, d);
77     m = m%n;
78     printf("\nOriginal Message Sent = %ld", m);
79
80     return 0;
81 }

```

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 یا Software 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 به انضمام کدهای نوشته شده و فایل‌های خروجی ابزار را به صورت فشرده، همراه با نام و شماره دانشجویی در سامانه درس‌افزار بارگذاری نمایید.

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

موفق باشید