

گزارش تکلیف ششم درس هم طراحی سخت افزار و نرم افزار

حسین قنبری

علیرضا مهدی برزی

۱ مقدمه

در این تکلیف کامپیوتری، یک سیستم کامل برای اجرای هم طراحی رمزنگاری به شیوه RSA را پیاده سازی کردیم. مهم ترین چالش ارتباط بین سخت افزار و نرم افزار بود که درباره نحوه پیاده سازی آن مفصل صحبت خواهیم کرد. در ضمن در انتهای گزارش تصاویر خروجی سیستم نهایی با استفاده از دستور gplatform به پیوست تقدیم می شود.

۲ شیوه پیاده سازی

با توجه به اینکه در صورت پروژه ذکر شده بود باید از شیوه memory-mapped برای برقراری ارتباط بین سخت افزار و نرم افزار استفاده کنیم؛ ما به همین شکل عمل کردیم و طبق مثال کتاب جلو رفتیم و بر همین اساس آدرس هایی از حافظه به صورت قراردادی انتخاب کردیم تا پل ارتباطی میان سخت افزار و نرم افزار باشد. ملاحظاتی در رابطه با شیوه تعریف هر پورت ارتباطی باید لحاظ می شد که شامل armsystemsink و armsystemsouce و core=myarm می شد؛ همچنین توجه به نوع پورت که ورودی است یا خروجی هم بسیار حائز اهمیت بود. در گام بعد هم، متغیرهایی volatile در سمت نرم افزار تعریف کردیم که با استفاده از اشاره گر به خانه های حافظه متناظر با هر بخش اشاره می کردند و فقط طراحی و پیاده سازی یک پروتکل ارتباطی لازم بود تا ارتباط سخت افزار و نرم افزار بطور کامل برقرار شود. در طراحی این پروتکل ارتباطی که بخشی از ماشین حالت اصلی سخت افزار بود باید به سخت افزار اطلاع می دادیم که چه موقع پارامترهای ورودی سیستم که در صورت تمرین مشخص شده بود آماده استفاده است و به همین خاطر یک پورت با نام REQ تعریف کردیم تا این وظیفه را برعهده بگیرد. بعد از خواندن پارامترهای مسئله از حافظه، لازم بود عملیات رمزگذاری و رمزگشایی به شیوه RSA انجام شود و نرم افزار می بایست تا زمان آماده شدن خروجی ها که شامل CipherText حاصل از رمزگذاری و PlainText حاصل از رمزگشایی بود صبر می کرد و برای همین یک پورت با نام ACK تعریف کردیم تا براساس وضعیتی که ماشین حالت سخت افزار تعیین می کند نرم افزار بصورت polling صبر کند تا نتایج آماده و در خانه های حافظه قرار بگیرد. برای سادگی بیشتر به تعداد پارامترهای موردنیاز که پیش از اجرای الگوریتم لازم بود در حافظه قرار بگیرد با offset های 8 تایی در حافظه مشترک بین سخت افزار و نرم افزار فضا تعریف کردیم که این فضا شامل محل قرارگیری متغیرهای ورودی و خروجی یعنی p و q و e و d و message و enc و dec بود. چالش مهم دیگر ماشین حالت اصلی سمت سخت افزار بود که آن را به اجزا کوچک تر تقسیم کردیم و مرحله به مرحله کارهای لازم را روی اطلاعات ورودی انجام دادیم. به همین ترتیب CFG های سمت سخت افزار را ذیل دسته های communication protocol, setter (data-proxy), intermediate calculations, encryption و decryption تقسیم کردیم که به وضوح در سورس نهایی قابل مشاهده می باشد. نکته ای که بسیار ما را اذیت کرد توجه به مقداره های همه پورت های خروجی در هر بار اجرای ماشین حالت بود که اگر این کار بدرستی انجام نمی شد دستور gplatform بطور کامل fail میشد و باعث عدم شبیه سازی می گردید که با بررسی نقطه به نقطه تمام سورس بالاخره توانستیم این مشکل را پیدا کنیم و بعضی از پورت های خروجی را از طریق دستور always مقداره ای کردیم و برخی دیگر را هم طی همه انتقال های صورت گرفته در ماشین حالت تا مشکل Missing Assignment برطرف شد.

برای تست سیستم از پارامترهای مختلفی استفاده کردیم که برای استخراج این پارامترها یک notebook برای نرم افزار mathematica نوشتیم تا اعداد اول رندم تولید و صحت پارامترهای لازم برای شبیه سازی نهایی را اعتبارسنجی کنیم. باتوجه به رفت و برگشت های زیاد

سورس کد تا برطرف شدن همه خطاها یک bash script هم برای ubuntu نوشتیم تا روند cross compile و اجرای gplatform با سرعت بیشتری انجام شود که شامل خطوط زیر بود:

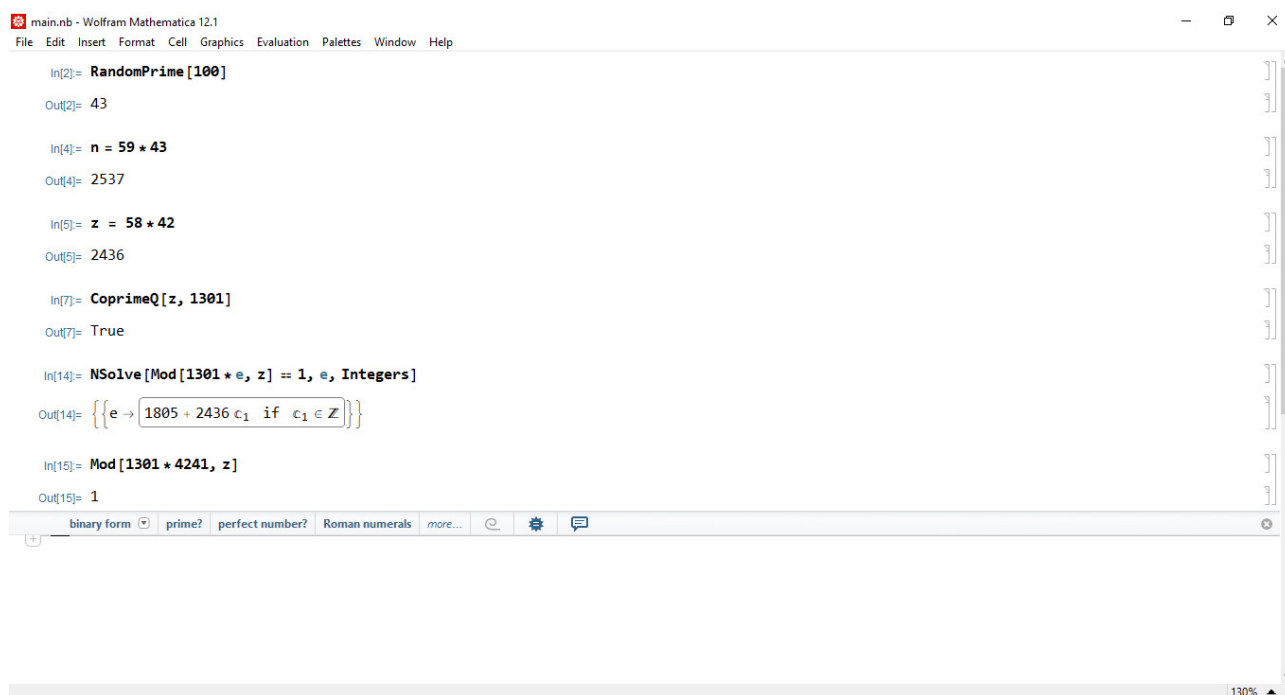
```
arm-linux-gcc -static rsa.c -o rsa
gplatform model.fdl
```

در پیاده‌سازی الگوریتم RSA همه قسمت‌های خواسته شده در سخت‌افزار پیاده‌سازی کردیم و فقط تامین ورودی اولیه و نمایش خروجی نهایی الگوریتم توسط نرم‌افزار صورت می‌پذیرد. بنابراین تمام حلقه‌های لازم چه برای رمزگذاری چه برای رمزگشایی بصورت ماشین حالت در سخت‌افزار پیاده‌سازی شده است و سعی کردیم با نام‌گذاری مناسب برای CFG ها پیچیدگی پیاده‌سازی الگوریتم را مدیریت کنیم. درضمن در روند پیکربندی gplatform ابتدا از کراس کامپایلرهای رسمی عرضه شده در apt-get استفاده کردیم که به هیچ نتیجه‌ای نتوانستیم برسیم و هیچ سازگاری با جزل نداشتند یعنی دستورات arm-linux-gnueabi-gcc و arm-linux-gnueabi-gcc gcc هیچکدام نتوانستند آبجکت‌فایل لازم برای شبیه‌سازی را در اختیار جزل قرار دهند که بعداً متوجه شدیم باید از arm-linux-gcc-3.2.deb استفاده می‌کردیم و به این ترتیب مشکل کراس کامپایل هم برطرف شد.

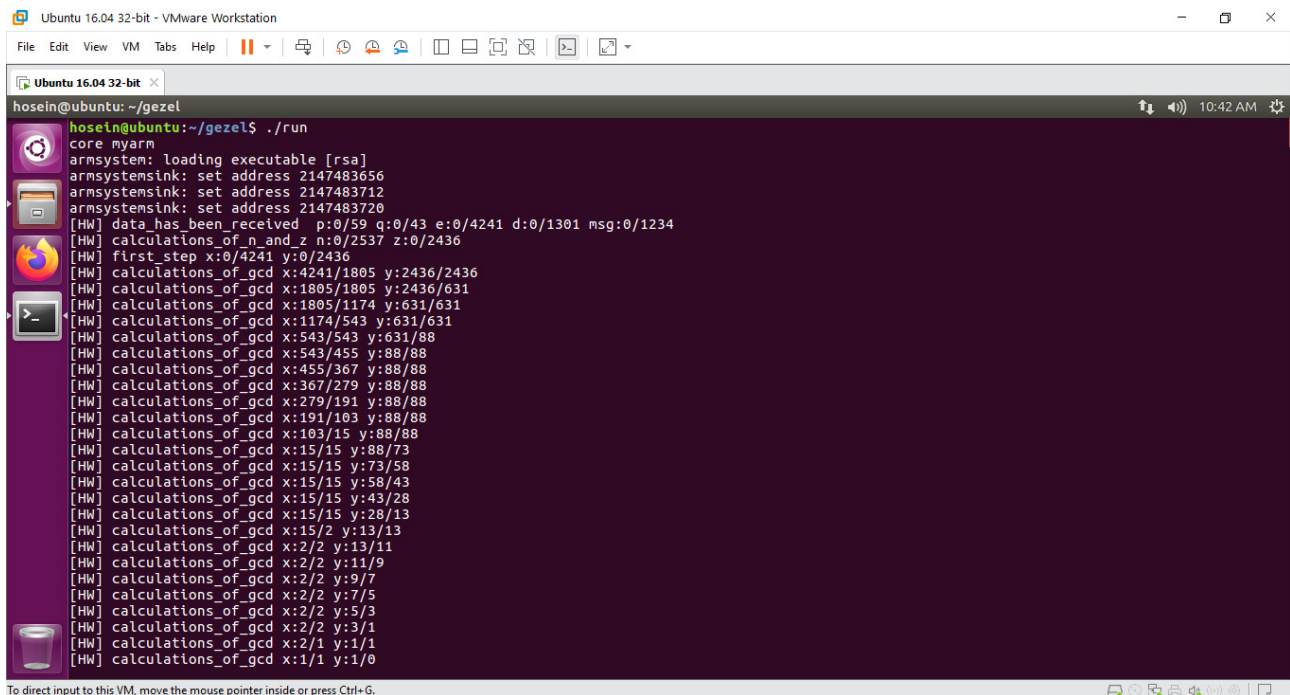
به پیوست این گزارش، فایل FDL سخت‌افزار، سورس زبان C نرم‌افزار، خروجی‌های VHDL جزل، فایل CrossCompile شده سورس، batch script کامپایل و اجرای شبیه‌سازی با gplatform و هم‌چنین notebook نرم‌افزار mathematica تقدیم می‌شود.

۳ خروجی‌های شبیه‌سازی با gplatform

تصویر ۱ – اجرای نوت‌بوک mathematica برای محاسبه p,q,e,d

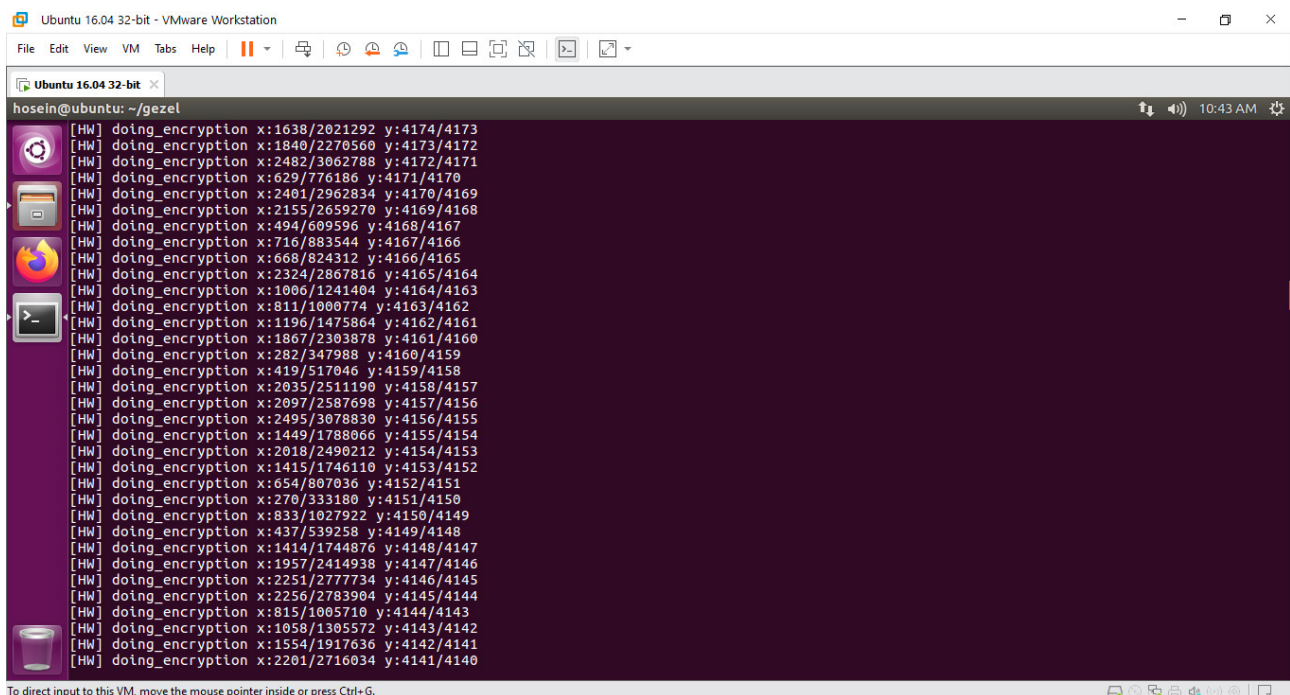


مجموعه تصاویر ۲ - اجرای دستور gplatform برای شبیه‌سازی تکلیف با پارامترهای بدست آمده از mathematica



```

hosein@ubuntu: ~/gezel
hosein@ubuntu:~/gezel$ ./run
core myarm
armsystem: loading executable [rsa]
armsystemsink: set address 2147483656
armsystemsink: set address 2147483712
armsystemsink: set address 2147483720
[HW] data has been received p:0/59 q:0/43 e:0/4241 d:0/1301 msg:0/1234
[HW] calculations_of_n_and_z n:0/2537 z:0/2436
[HW] first_step x:0/4241 y:0/2436
[HW] calculations_of_gcd x:4241/1805 y:2436/2436
[HW] calculations_of_gcd x:1805/1805 y:2436/631
[HW] calculations_of_gcd x:1805/1174 y:631/631
[HW] calculations_of_gcd x:1174/543 y:631/631
[HW] calculations_of_gcd x:543/543 y:631/88
[HW] calculations_of_gcd x:543/455 y:88/88
[HW] calculations_of_gcd x:455/367 y:88/88
[HW] calculations_of_gcd x:367/279 y:88/88
[HW] calculations_of_gcd x:279/191 y:88/88
[HW] calculations_of_gcd x:191/103 y:88/88
[HW] calculations_of_gcd x:103/15 y:88/88
[HW] calculations_of_gcd x:15/15 y:88/73
[HW] calculations_of_gcd x:15/15 y:73/58
[HW] calculations_of_gcd x:15/15 y:58/43
[HW] calculations_of_gcd x:15/15 y:43/28
[HW] calculations_of_gcd x:15/15 y:28/13
[HW] calculations_of_gcd x:15/2 y:13/13
[HW] calculations_of_gcd x:2/2 y:13/11
[HW] calculations_of_gcd x:2/2 y:11/9
[HW] calculations_of_gcd x:2/2 y:9/7
[HW] calculations_of_gcd x:2/2 y:7/5
[HW] calculations_of_gcd x:2/2 y:5/3
[HW] calculations_of_gcd x:2/2 y:3/1
[HW] calculations_of_gcd x:2/1 y:1/1
[HW] calculations_of_gcd x:1/1 y:1/0
  
```



```

hosein@ubuntu: ~/gezel
hosein@ubuntu:~/gezel$ ./run
[HW] doing_encryption x:1638/2021292 y:4174/4173
[HW] doing_encryption x:1840/2270560 y:4173/4172
[HW] doing_encryption x:2482/3062788 y:4172/4171
[HW] doing_encryption x:629/776186 y:4171/4170
[HW] doing_encryption x:2401/2962834 y:4170/4169
[HW] doing_encryption x:2155/2659270 y:4169/4168
[HW] doing_encryption x:494/609596 y:4168/4167
[HW] doing_encryption x:716/883544 y:4167/4166
[HW] doing_encryption x:668/824312 y:4166/4165
[HW] doing_encryption x:2324/2867816 y:4165/4164
[HW] doing_encryption x:1006/1241404 y:4164/4163
[HW] doing_encryption x:811/1000774 y:4163/4162
[HW] doing_encryption x:1196/1475864 y:4162/4161
[HW] doing_encryption x:1867/2303878 y:4161/4160
[HW] doing_encryption x:282/347988 y:4160/4159
[HW] doing_encryption x:419/517046 y:4159/4158
[HW] doing_encryption x:2035/2511190 y:4158/4157
[HW] doing_encryption x:2097/2587698 y:4157/4156
[HW] doing_encryption x:2495/3078830 y:4156/4155
[HW] doing_encryption x:1449/1788066 y:4155/4154
[HW] doing_encryption x:2018/2490212 y:4154/4153
[HW] doing_encryption x:1415/1746110 y:4153/4152
[HW] doing_encryption x:654/807036 y:4152/4151
[HW] doing_encryption x:270/333180 y:4151/4150
[HW] doing_encryption x:833/1027922 y:4150/4149
[HW] doing_encryption x:437/539258 y:4149/4148
[HW] doing_encryption x:1414/1744876 y:4148/4147
[HW] doing_encryption x:1957/2414938 y:4147/4146
[HW] doing_encryption x:2251/2777734 y:4146/4145
[HW] doing_encryption x:2256/2783904 y:4145/4144
[HW] doing_encryption x:815/1005710 y:4144/4143
[HW] doing_encryption x:1058/1305572 y:4143/4142
[HW] doing_encryption x:1554/1917636 y:4142/4141
[HW] doing_encryption x:2201/2716034 y:4141/4140
  
```

Ubuntu 16.04 32-bit - VMware Workstation

File Edit View VM Tabs Help

hosein@ubuntu: ~/gezel

```
[HM] doing_encryption x:36/44424 y:2280/2279
[HM] doing_encryption x:1295/1598030 y:2279/2278
[HM] doing_encryption x:2257/2785138 y:2278/2277
[HM] doing_encryption x:2049/2528466 y:2277/2276
[HM] doing_encryption x:1614/1991676 y:2276/2275
[HM] doing_encryption x:131/161654 y:2275/2274
[HM] doing_encryption x:1823/2249582 y:2274/2273
[HM] doing_encryption x:1800/2221200 y:2273/2272
[HM] doing_encryption x:1325/1635050 y:2272/2271
[HM] doing_encryption x:1222/1507948 y:2271/2270
[HM] doing_encryption x:970/1196980 y:2270/2269
[HM] doing_encryption x:2053/2533402 y:2269/2268
[HM] doing_encryption x:1476/1821384 y:2268/2267
[HM] doing_encryption x:2355/2906070 y:2267/2266
[HM] doing_encryption x:1205/1486970 y:2266/2265
[HM] doing_encryption x:288/355392 y:2265/2264
[HM] doing_encryption x:212/261608 y:2264/2263
[HM] doing_encryption x:297/366498 y:2263/2262
[HM] doing_encryption x:1170/1443780 y:2262/2261
[HM] doing_encryption x:227/280118 y:2261/2260
[HM] doing_encryption x:1048/1293232 y:2260/2259
[HM] doing_encryption x:1899/2343366 y:2259/2258
[HM] doing_encryption x:1715/2116310 y:2258/2257
[HM] doing_encryption x:452/557768 y:2257/2256
[HM] doing_encryption x:2165/2671610 y:2256/2255
[HM] doing_encryption x:149/183866 y:2255/2254
[HM] doing_encryption x:1202/1483268 y:2254/2253
[HM] doing_encryption x:1660/2048440 y:2253/2252
[HM] doing_encryption x:1081/1333954 y:2252/2251
[HM] doing_encryption x:2029/2503786 y:2251/2250
[HM] doing_encryption x:2304/2843136 y:2250/2249
[HM] doing_encryption x:1696/2092864 y:2249/2248
[HM] doing_encryption x:2376/2931984 y:2248/2247
[HM] doing_encryption x:1749/2158266 y:2247/2246
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Ubuntu 16.04 32-bit - VMware Workstation

File Edit View VM Tabs Help

hosein@ubuntu: ~/gezel

```
[HM] doing_decryption x:2227/1031101 y:1115/1114
[HM] doing_decryption x:1079/499577 y:1114/1113
[HM] doing_decryption x:2325/1076475 y:1113/1112
[HM] doing_decryption x:787/364381 y:1112/1111
[HM] doing_decryption x:1590/736170 y:1111/1110
[HM] doing_decryption x:440/203720 y:1110/1109
[HM] doing_decryption x:760/351880 y:1109/1108
[HM] doing_decryption x:1774/821362 y:1108/1107
[HM] doing_decryption x:1911/884793 y:1107/1106
[HM] doing_decryption x:1917/887571 y:1106/1105
[HM] doing_decryption x:2158/999154 y:1105/1104
[HM] doing_decryption x:2113/978310 y:1104/1103
[HM] doing_decryption x:1574/728762 y:1103/1102
[HM] doing_decryption x:643/297789 y:1102/1101
[HM] doing_decryption x:880/407440 y:1101/1100
[HM] doing_decryption x:1520/703760 y:1100/1099
[HM] doing_decryption x:1011/468093 y:1099/1098
[HM] doing_decryption x:1285/594955 y:1098/1097
[HM] doing_decryption x:1297/600511 y:1097/1096
[HM] doing_decryption x:1779/823677 y:1096/1095
[HM] doing_decryption x:1689/782007 y:1095/1094
[HM] doing_decryption x:611/282893 y:1094/1093
[HM] doing_decryption x:1286/595418 y:1093/1092
[HM] doing_decryption x:1760/814880 y:1092/1091
[HM] doing_decryption x:503/232889 y:1091/1090
[HM] doing_decryption x:2022/936186 y:1090/1089
[HM] doing_decryption x:33/15279 y:1089/1088
[HM] doing_decryption x:57/26391 y:1088/1087
[HM] doing_decryption x:1021/472723 y:1087/1086
[HM] doing_decryption x:841/389383 y:1086/1085
[HM] doing_decryption x:1222/565786 y:1085/1084
[HM] doing_decryption x:35/16205 y:1084/1083
[HM] doing_decryption x:983/455129 y:1083/1082
[HM] doing_decryption x:1006/465778 y:1082/1081
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

```
hosein@ubuntu: ~/gezel
[HM] doing_decryption x:2307/1068141 y:29/28
[HM] doing_decryption x:64/29632 y:28/27
[HM] doing_decryption x:1725/798675 y:27/26
[HM] doing_decryption x:2057/952391 y:26/25
[HM] doing_decryption x:1016/470408 y:25/24
[HM] doing_decryption x:1063/492169 y:24/23
[HM] doing_decryption x:2528/1170464 y:23/22
[HM] doing_decryption x:907/419941 y:22/21
[HM] doing_decryption x:1336/618568 y:21/20
[HM] doing_decryption x:2077/961651 y:20/19
[HM] doing_decryption x:128/59264 y:19/18
[HM] doing_decryption x:913/422719 y:18/17
[HM] doing_decryption x:1577/730151 y:17/16
[HM] doing_decryption x:2032/940816 y:16/15
[HM] doing_decryption x:2126/984338 y:15/14
[HM] doing_decryption x:2519/1166297 y:14/13
[HM] doing_decryption x:1814/839882 y:13/12
[HM] doing_decryption x:135/62505 y:12/11
[HM] doing_decryption x:1617/748671 y:11/10
[HM] doing_decryption x:256/118528 y:10/9
[HM] doing_decryption x:1826/845438 y:9/8
[HM] doing_decryption x:617/285671 y:8/7
[HM] doing_decryption x:1527/707001 y:7/6
[HM] doing_decryption x:1715/794045 y:6/5
[HM] doing_decryption x:2501/1157963 y:5/4
[HM] doing_decryption x:1091/505133 y:4/3
[HM] doing_decryption x:270/125010 y:3/2
[HM] doing_decryption x:697/322711 y:2/1
[HM] doing_decryption x:512/237056 y:1/0
[HM] doing_decryption x:1115/516245 y:0/4294967295
[SOFTWARE] encrypted data fetched using proxy = 463
[SOFTWARE] decrypted data fetched using proxy = 1234
Total Cycles: 5872316
hosein@ubuntu:~/gezel$
```

۴ نتیجه گیری

اجرای این پروژه یک تجربه جدید بود؛ به این خاطر که توانستیم در محیط شبیه‌سازی ارتباط بین سخت‌افزار و نرم‌افزار برای انجام یک مجموعه از محاسبات را از طریق به اشتراک‌گذاری فضای memory انجام دهیم. هرچند که ابزار جزل به خاطر صنعتی نبودن محدودیت‌های فراوانی دارد و واقعا برای خطاهای زمان اجرای آن مستندات دقیقی وجود ندارد اما برای تست چنین سناریوهایی بسیار خوب ظاهر می‌شود. البته اگر ابزار تجاری در اختیار داشتیم قطعا خروجی‌های باکیفیت تری می‌توانستیم با سرعت بیشتر تولید و پیاده‌سازی کنیم. طراحی سخت‌افزارهای بهینه رمزنگاری قطعا یکی از مهم‌ترین حوزه‌های طراحی سخت‌افزار می‌باشد که براساس تجربیاتی که از پیاده‌سازی این تکلیف کامپیوتری بدست آوردیم می‌توانیم تست ایده‌های مختلف در این حوزه را با شبیه‌ساز جزل انجام دهیم. شایان ذکر است به خاطر اینکه اعداد اول نسبتاً بزرگی برای شبیه‌سازی انتخاب شد نتیجه نهایی الگوریتم در زمان نسبتاً طولانی محاسبه می‌شد و شاید بتوانیم با محول کردن برخی از محاسبات رایج و ابتدایی به نرم‌افزار سرعت کلی سیستم را ارتقا دهیم.