هوالحق

تمرین کامپیوتری شماره ۱
امنیت شبکه، بهار ۹۷

در این تمرین قصد داریم یک نسخه شخصی از RSA را پیاده سازی کنیم. به پیاده سازی RSA مطابق آنچه در منابع آموزشی به آن پرداخته می‌شود، Textbook RSA می‌گویند. کد تحویلی شما می‌بایست شامل ۴ فایل مجزا به شرح زیر باشد:

۱) فایل Keygen

- در این قسمت شما می‌بایست کلیدهای خصوصی و عمومی را تولید نمایید. برنامه شما در ابتدای اجرا، دو عدد اول P و Q را از کاربر تقاضا می‌کند. برنامه باید اول بودن هر دوی این اعداد را بررسی کرده و برابر نبودن آن‌ها را احراز کند. در صورت عدم برقراری شرایط، با پیغام خطای مناسب اجرای برنامه متوقف می‌شود. همچنین با توجه به نوع رمزنگاری (که در قسمت‌های بعد شرح داده خواهد شد)، مقدار پیمانه n (برابر حاصلضرب P و Q) بایستی بزرگتر مساوی ۲۵۶ بوده و در غیر اینصورت از اجرای برنامه ممانعت به عمل آید.
- پس از این که مقدار P و Q مناسب توسط برنامه دریافت شد، برنامه به صورت خودکار یک عدد تصادفی e مطابق با شرایط الگوریتم RSA تولید می‌کند. در این مرحله، توجه به جزئیات شروط لازم برای e مورد نیاز است. برنامه شما باید در تمامی فضای حالت به صورت تصادفی جستجو کرده و یک e مطلوب و تصادفی را بیابد.
- پس از یافتن یک e مطلوب، برنامه شما باید به تولید d مبادرت ورزد. یافتن d باید صرفاً توسط الگوریتم اقلیدسی انجام گیرد و پیاده سازی آن ذیل یک تابع الزامی است. در این مرحله برای پیدا کردن d از bruteforce استفاده نکنید.
- در انتها، برنامه شما باید به ترتیب مقادیر P، Q، n، Phi(n)، e و d را در خروجی کنسول چاپ نماید.
- کلید خصوصی شامل زوج مقادیر d و n می‌باشد. برنامه شما در انتهای اجرا می‌بایست یک فایل با نام PRIVATE.key تولید کرده که در خط اول آن مقدار d و در خط دوم، مقدار n چاپ شده باشد.
- به همین ترتیب، کلید عمومی نیز در یک فایل با نام PUBLIC.key که خط اول آن شامل مقدار e بوده و خط دوم شامل n است باید درج شود.

۲) فایل Encrypt

- این فایل به انجام رمزنگاری روی فایل ورودی با نام input.txt پرداخته و با استفاده از کلید عمومی موجود در فایل PUBLIC.key عملیات رمزنگاری را انجام می‌دهد.
- با توجه به محدودیت طول پیمانه در الگوریتم RSA و طول زیاد فایل‌های ورودی، نیاز است تا رمزنگاری در این قسمت به صورت بلوکی انجام گیرد. طول هر بلوک رمزنگاری ۱ بایت (معادل ۸ بیت) بوده و عملیات رمز به ازای هر یک بایت صورت خواهد گرفت.
- طبق مباحث الگوریتم رمزنگاری RSA، اگر طول پیغام M بوده و طول پیمانه نیز n باشد (n > M)، طول C لزوماً به اندازه M نیست؛ ولی از n کمتر خواهد بود. این خصوصیت الگوریتم RSA باعث می‌شود که عملیات بلوکه سازی فایل ورودی به راحتی رمزنگارهای بلوکی مانند DES و یا AES نباشد، زیرا طول خروجی یک بلوک رمز RSA برابر ورودی آن نبوده و حتی بلوک‌های متوالی، خروجی‌هایی با طول‌های یکسان نیز ایجاد نخواهند کرد. برای حل این مشکل، تمام

هوالحق

تمرین کامپیوتری شماره ۱
امنیت شبکه، بهار ۹۷

بلوک‌های خروجی فرآیند رمزنگاری فایل را با طول یکسان و برابر ماکسیمم طول ممکن در ایجاد یک بلوک C در نظر می‌گیریم.

- یک پیغام ۸ بیتی M ماکسیمم مقدار ۲۵۵ را می‌تواند اتخاذ کند. فلذا با توجه به شرط $n > M$، مقدار n می‌بایست حداقل برابر ۲۵۶ باشد تا در فرآیند رمزنگاری هر بلوک هشت بیتی با مقدار دلخواه، مشکلی ایجاد نشود. به این ترتیب، در صورتی که n بزرگتر مساوی ۲۵۶ باشد، مقدار C نیز می‌تواند مقداری از عدد ۰ تا n-۱ باشد. این بدین معنیست که ماکسیمم طول C برابر تعداد بیت‌های لازم برای نگه‌داری عدد n-۱ می‌باشد.
- با توجه به اینکه واحدهای تشکیل دهنده یک فایل کمتر از ۱ بایت نمی‌توانند باشند، مقادیر ذکر شده می‌بایست به نزدیک ترین ضریب بایت (هشت تایی) کران بالا گرد شود. با توجه به این نکته، در صورتی که n برابر ۲۵۶ باشد، ماکسیمم ظرفیت مورد نیاز برای هر بلوک C برابر ۱ بایت؛ در صورتی که n بزرگتر از ۲۵۶ و کوچکتر مساوی ۶۵,۵۳۶ باشد ماکسیمم سایز بلوک C برابر ۲ بایت، بزرگتر از ۶۵,۵۳۶ و کوچکتر مساوی ۱۶,۷۷۷,۲۱۶ باشد ماکسیمم ۳ بایت و به همین ترتیب.
- برنامه رمزنگار می‌بایست با بررسی مقدار n موجود در کلید عمومی، مقدار بایت مورد نیاز برای بلوک‌های C را بدست بیاورد. پس از این، هر بلوک ۱ بایتی از فایل ورودی را رمز کرده و حاصل رمز را (که می‌تواند ۱ بایت، ۲ بایت، ۳ بایت و یا بیشتر بسته به مقدار n باشد) در فایل خروجی به نام input.txt.enc ذخیره کند. در این صورت، فایل input.enc سایز یک برابر، دو برابر، سه برابر و یا بیشتر (بسته به مقدار n) نسبت به فایل اصلی پیدا خواهد کرد.

۳) فایل Decrypt

- این فایل با استفاده از کلید خصوصی موجود در فایل PRIVATE.key به رمزگشایی یک فایل ورودی با نام input.txt.enc می‌پردازد. این برنامه در ابتدا محتوای کلید خصوصی را قرائت کرده و بسته به مقدار n، تعداد بایت‌هایی که می‌بایست به عنوان یک بلوک C در نظر بگیرد را بدست می‌آورد. این مقدار می‌تواند برابر ۱ بایت، ۲ بایت، ۳ بایت و یا بیشتر بسته به مقدار n خواهد بود.
- پس از این، در هر نوبت به خواندن C با سایز مورد نظر از فایل ورودی کرده و با انجام عملیات رمزگشایی RSA روی آن، M را که یک مقدار ۱ بایتی بوده بدست آورده و آن را در فایلی با نام output.txt ذخیره می‌نماید. فایل output.txt می‌بایست کاملاً مشابه input.txt بدست بیاید.

۴) فایل Crack

- در این برنامه، به شکستن رمز یک فایل ورودی encrypted.txt.enc پرداخته می‌شود. این برنامه باید فایل را به طور کامل رمزگشایی کرده و همچنین مقدار دقیق n و d را بدست آورده و در انتهای عملیات در خروجی چاپ کند. فایل رمزگشایی شده نیز می‌بایست با نام cracked.txt.enc ایجاد گردد.
- مفروض است که فایل encrypted.txt.enc، رمز شده‌ی یک فایل تماماً متنی بوده که شامل انواع کاراکترهای بصری و همچنین فاصله، tab و newline می‌باشد و شامل مواردی به جز این نبوده است.
- همچنین می‌دانیم فایلی که با نام (YOUR_SID).txt.enc در اختیار شما قرار داده شده است، پیمانه‌ی رمزی برابر n که مابین ۲۵۶ و ۵۱۲ بوده است را دارد (و C ها ۲ بایتی هستند).

هوالحق

تمرین کامپیوتری شماره ۱
امنیت شبکه، بهار ۹۷

نکات

* برنامه‌های نوشته شده می‌بایست Robustness لازم را دارا باشد و در هر مرحله با پیغام خطای مناسب علت عدم اجرای صحیح را اطلاع رسانی کند. تحت هیچ شرایطی بروز Segmentation Fault و یا Exception ها و Error های پیش فرض در روند اجرای برنامه قابل قبول نیست.

* تمامی الگوریتم‌های لازم می‌بایست به صورت کامل پیاده سازی شوند. تنها استفاده از توابع ریاضی توان، لگاریتم و اپراتورهای ریاضی پیش‌فرض قابل قبول است.

* توجه خود را به ظرفیت data structure زبان مربوطه جلب کنید تا overflow رخ ندهد. متغیرهای برنامه شما می‌بایست تحمل اجرای برنامه ای با پیمانه n ای کوچکتر مساوی ۴،۲۹۴،۹۶۷،۲۹۶ را داشته باشد (ماکسیمم ۴ بایت).

* برنامه شما می‌بایست با یکی از زبان‌های C، C++ و یا Python3 نوشته شده باشد و همچنین در محیط Linux قابلیت کامپایل/تفسیر/اجرا را داشته باشد.

* در این تمرین، تأکیدی روی Performance نیست. تمرکز خود را روی پیاده سازی دقیق الگوریتم قرار دهید.

* ملاک جزئیات الگوریتم‌ها، مطالب تدریس شده در کلاس درس می‌باشد.

* ذخیره سازی بایت های رمز شده در فایل باید به صورت Big Endian باشد (بایت های ارزش بیشتر زودتر، در سمت چپ، درج شوند).

* با توجه به تصحیح خودکار تمرین‌ها، لطفاً توجه لازم را به نام‌گذاری فایل‌ها، ورودی ها و خروجی ها مبذول دارید.

* به شماتیک ضمیمه این تمرین توجه لازم را داشته باشید.

* برنامه شما باید علاوه بر رمزنگاری و رمزگشایی فایل‌های متنی، توانایی انجام عملیات روی هر نوع فایلی را داشته باشد. این امر با برآورد شرط مشابهت صد درصدی بایت به بایت فایل‌های input.txt و output.txt تضمین خواهد شد؛ فلذا با تغییر نام هر گونه فایل متنی، مالتی‌مدیا و غیره به نام‌های مربوطه، انجام عملیات رمزنگاری و بازگرداندن پسوند آن‌ها پس از رمزگشایی، فایل‌ها بایستی مشابهت کامل و هویت خود را حفظ نمایند.

* فایل‌های تحویلی شما فقط و فقط شامل **۴ فایل سورس کد Keygen، Encrypt، Decrypt، Crack** و همچنین **یک فایل متنی شامل پیغام شکسته شده، مقدار d و مقدار n فایل رمز شده** خواهد بود.