تمرین عملی ۱

نویسندگان: محمدرضا مومنی – سید محمد مهدی نیکنام باقری

در این تمرین پیاده سازی که انجام شده مربوط به یک برنامه چت ساده هست که در ارسال پیام از رمزگذاری های متقارن، نا متقارن و همچنین هش برای اصالت سنجی پیام استفاده می کند.

مهمترین کتابخانه های استفاده شده در این برنامه عبارت اند از:

- Cryptography
- RSA
- Hashlib

Cryptography

این کتابخانه برای پیاده سازی رمز متقارن مورد استفاده قرار می گیرد. این کتابخانه علاوه بر امکان استفاده برای رمز متقارن امکاناتی را برای رمزگذاری نامتقارن فراهم می کند. همچنین امکان ایجاد عدد تصادفی به صورت امن را فراهم می کند. از قابلیت های دیگر این کتابخانه امکان استفاده برای هش کردن داده ها و رمز های عبور را دارد.

RSA

این کتابخانه امکان استفاده از رمز نامتقارن RSA را فراهم می کند. از ویژگی هایی که این کتابخانه دارد امکان ایجاد کلید عمومی و خصوصی، رمز کردن و بازیابی داده رمز شده می باشد. همچنین امکانی را برای امضای دیجیتال و خروجی گرفتن و یا لود کردن کلید ها فراهم می کند.

Hashlib

این کتابخانه به منظور هش کردن داده استفاده میشود و همچنین امکان اضافه کردن salt به داده هش شده را فراهم می کند.

پروژه

پروژه انجام شده شامل فایل های زیر می باشد:

- Srever.py
- Client.py
- Hash.py
- Asymmetric.py
- Symmetric

Server.py و client.py

متد های موجود در دو فایل:

- exchange_key()
- receive ()
- send()

این دو فایل شامل سه متد می باشد که برای تبادل کلید متقارن، ارسال و دریافت داده می باشد. متد اول که برای تبادل کلید متقارن استفاده می شود به این صورت عمل می کند که ابتدا پس از برقراری ارتباط با client، کلید عمومی را از client دریافت می کند و به وسیله ان کلید متقارن را برای client ارسال می کند. تبادل کاید بلافاصله پس از اجرای برنامه انجام میشود.

بعد از تبادل کلید دو نخ به صورت همزمان شروع به اجرا می کنند. یکی به منظور اجرای متد ارسال داده از سمت سرور به کلاینت و دیگری برای دریافت داده از سمت کلاینت و به طور مشابه در سمت کلاینت یک متد برای دریافت داده از سمت سرور و دیگری برای ارسال داده. در متد send در server.py، داده ای از کاربر دریافت می شود سپس هش آن با کمک کلید عمومی کلاینت رمز شده و ارسال می شود و همچنین داده دریافت شده با کمکم رمز متقارن رمز شده و به سمت کلاینت ارسال می شود. در سمت کلاینت در متد receive، هش داده و

خود داده دریافت شده و رمزگشایی می شوند. بعد از آن با استفاده از تابع validator در فایل hash.py ، هش دریافت شده و پیام اصالت سنجی می شوند.

در متد send در فایل client.py ، داده صرفا با رمز متقارن رمز شده و ارسال می شود.

Hash.py

متد های این فایل:

- validator(data, hashed)
- hashing(data)

متد validator وظیفه بررسی هش داده دریافتی را دارد تا از اصالت دریافت پیام اطمینان حاصل شود. به این صورت که ابتدا داده دریافتی را هش می کند و خروجی را با هش دریافتی از سرور بررسی می کند در صورت یکسان بودن هش ها نتیجه می شود که داده دارای اصالت می باشد

متد hashing ، وظیفه هش کردن داده ها را بر عهده دارد.

Symmetric.py

متد های این فایل:

- encode(data, key)
- decode(data, key)

متد encode وظیفه رمز داده ها به وسیله رمز متقارن را بر عهده دارد.

متد decode داده دریافتی را با کمک رمز متقارن، رمزگشایی می کند.

Assymetric.py

متد های این فایل:

- encrypt(data, u_key)
- decrypt(data, r_key)

متد encrypt داده دریافتی را به وسیله کلید عمومی رمز می کند.

متد decrypt داده دربافتی را به کمک کلید خصوصی رمز گشایی می کند.