پروژه امنیت داده و شبکه

پیام‌رسان امن

استاد
دکتر مرتضی امینی

نویسندگان
محمدرضا مفیضی - ۹۸۱۰۶۰۵۹
رضا علیپور - ۹۸۱۰۵۹۳۲
محمدعلی کاکاوند - ۹۸۱۰۲۱۱۹

دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

بهار ۱۴۰۲

فهرست مطالب

۱ مقدمه

در این نوشته، سعی می‌کنیم توضیحاتی درباره نحوه پیاده‌سازی پیام‌رسان امن با ویژگی‌های خواسته‌شده در سند پروژه ارائه دهیم. این پیام‌رسان از رمزنگاری انتها به انتها برای تبادل پیام استفاده می‌کند و به صورت کلاینت-سروری پیاده‌سازی شده است. پیام‌های رد و بدل شده میان هر کلاینت و سرور نیز با استفاده از کلید مشترک بین آن‌ها رمز خواهد شد.

برای پیاده‌سازی بخش سرور از چارچوب Django استفاده شده است و درخواست‌ها از طریق پروتکل Rest به سرور ارسال می‌شود. برای ارسال پیام از سرور به کلاینت نیز از پروتکل WebSocket استفاده می‌شود. جزئیات رمزگذاری انتها به انتها و پیاده‌سازی سرور و کلاینت در ادامه آمده است.

۲ کلاینت

۱.۲ کلیات

در سمت کلاینت جفت‌کلید خصوصی و عمومی او برای رمزنگاری نامتقارن جهت ایجاد نشست با سرور و یا کلاینت دیگر نگه‌داری می‌شود. برای ارسال پیام به سرور فقط از post استفاده می‌شود و همواره توکنی که سرور برای نشست به ما داده است در کنار پیام قرار گرفته و همگی با کلید نشست با سرور رمز می‌شوند.

۲.۲ ایجاد حساب کاربری

برای ایجاد و وارد شدن به حساب کاربری، کلاینت باید نام کاربری و رمز عبور را از او دریافت کند. سپس این اطلاعات برای ثبت‌نام به سرور فرستاده می‌شود. همچنین در صورتی که ثبت‌نام با موفقیت انجام شد، کاربر کلید عمومی خود به همراه پیش‌کلید امضاشده و امضای آن و یک لیست از پیش‌کلیدهای یک‌بار مصرف را برای سرور ارسال می‌کند.

ورود به حساب کاربری نیز با گرفتن نام کاربری و رمز عبور انجام می‌شود. در پاسخ یک توکن به کاربر داده می‌شود تا در درخواست‌های بعدی از آن استفاده شود.

۳.۲ ارسال و دریافت پیام

برای ارسال پیام به یک کاربر و رمزگذاری انتها به انتهای آن به این صورت عمل می‌کنیم.

برای ارسال پیام از پروتکل Double Ratchet استفاده می‌کنیم. این پروتکل ویژگی‌های محرمانگی پیش‌رو و پس‌رو را برای ما فراهم می‌کند. جزئیات بیشتر این پروتکل در این لینک آمده است.

در صورتی که کاربر دیگر آفلاین باشد از الگوریتم 3DH استفاده می‌شود. به این صورت که ابتدا ۳ کلید `key identity`، `signed prekey` و `signature prekey` کاربر آفلاین را (که در هنگام ثبت‌نام با سرور به اشتراک گذاشته است) از سرور دریافت می‌کنیم. سپس سه بار با کلیدهای دریافت‌شده DH می‌زنیم:

$$DH1 = DH(IK_A, SPK_B)$$
$$DH2 = DH(EK_A, IK_B)$$
$$DH3 = DH(EK_A, SPK_B)$$
$$DH4 = DH(EKA, OPKB)$$
$$SK = KDF(DH1||DH2||DH3||DH4)$$

درنهایت به یک کلید خواهیم رسید که پیام را با استفاده از آن رمز و به سرور ارسال می‌کنیم. سرور نیز هنگام آنلاین‌شدن فرد مقابل پیام رمزشده را به او ارسال می‌کند.

به دلیل استفاده‌شدن از `one-time prekey` امکان اجرای حمله تکرار نخواهد بود. همچنین به دلیل استفاده از امضا روی پیام‌های امکان انکار وجود نخواهد داشت.

جزئیات بیشتر این روش در این لینک قابل مشاهده است.

۴.۲ نگه‌داری پیام‌ها به‌صورت امن

پیام‌های هر کاربر با کاربران دیگر توسط کلیدی که توسط کلید مشتق‌شده از رمز عبور فرد ساخته می‌شود رمز می‌شوند. با ورود کاربر به حساب کاربری خود می‌توان پیام‌های قبلی را مشاهده کرد.

۵.۲ نگه‌داری کلید به‌صورت امن

کلیدهای هر کاربر با کاربران دیگر نیز با استفاده از کلید گفته‌شده در بخش قبلی رمز می‌شوند.

۶.۲ ایجاد و مدیریت گروه

برای ایجاد یک گروه کاربر درخواست خود را به همراه نام گروه به سرور ارسال می‌کند.
برای ساخت کلید گروه از الگوریتم AES برای ساخت یک کلید نشست متقارن استفاده می‌شود.
کاربر به صورت پیش‌فرض ادمین گروه خواهد بود و می‌تواند کاربران آنلاین دیگر را با ارسال نام کاربری آن‌ها به سرور به گروه اضافه کند. بعد از اضافه‌شدن کاربر لازم است تا کلید نشست گروه با او به اشتراک گذاشته شود.
برای تبادل کلید با کاربر جدید ابتدا با استفاده از پروتکل DH یک کلید موقت برای انتقال کلید استفاده می‌شود. سپس کلید نشست اصلی از طریق این کلید یک‌بار مصرف با کاربر جدید داده می‌شود.

۷.۲ تایید صحت نشست

برای تایید صحت نشست کاربران می‌توانند با انتخاب گزینه مورد نظر ، چکیده بخشی از کلید نشست را به صورت شکلک‌هایی مشاهده کنند و با مقایسه آن در دو سمت اتصال از امن بودن ارتباط اطمینان حاصل کنند.

۸.۲ تازه‌سازی کلیدهای نشست

هر کاربر می‌تواند با انتخاب گزینه‌ای کلید نشست جدیدی را برای ارتباط تولید کند و بقیه پیام‌ها با کلید جدید رمز خواهند شد. هنگام حذف یک کاربر از گروه نیز همین اتفاق خواهد افتاد و با تبادل کلید جدید نشست بروزرسانی می‌شود.

۳ سرور

۱.۳ کلیات

سرور فقط درخواست‌های post را می‌پذیرد و برای ارتباط با کاربران از کلید نشست تبادل‌شده استفاده می‌کند. همچنین سرور دسترسی به پیام‌های انتها به انتهای دو کاربر که با کلید نشست بین آن دو رمز شده است ندارد. برای هر درخواست از سمت کلاینت نیز، سرور توکن داده‌شده را بررسی می‌کند و در صورت صحت درخواست را انجام می‌دهد.

۲.۳ ایجاد حساب کاربری

در سمت سرور به هنگام دریافت نام کاربری و رمز عبور، یک salt به صورت تصادفی ایجاد می‌شود و چکیده رمز با salt در پایگاه‌داده ذخیره می‌شود. با این کار حتی در صورت حمله به پایگاه‌داده نیز رمزهای کاربران لو نمی‌رود. هنگام لاگین نیز سرور رمز داده‌شده را کنار salt ذخیره‌شده می‌گذارد و با مقدار ذخیره‌شده در پایگاه داده مقایسه می‌کند.

۳.۳ نمایش کاربران آنلاین

برای نمایش کاربران آنلاین کاربرهایی را که اتصال WebSocket فعال دارند به کاربر ارسال می‌کنیم.

۴.۳ ایجاد و مدیریت گروه

ایجاد گروه با درخواست کاربر به سرور انجام می‌شود. سرور اطلاعات گروه‌ها به همراه کاربران و نقش‌ها آن‌ها را ذخیره می‌کند.

۵.۳ ارسال پیام‌ها به مقصد

سرور با دریافت پیام از سمت یک کاربر و نام کاربری فرد مقصد آن را با استفاده از اتصال WebSocket به مقصد ارسال می‌کند. سرور پیام دریافت‌شده را در صورتی که کاربر دیگر آنلاین باشد در سمت خود ذخیره نخواهد کرد.

۴ نیازمندی‌های امنیتی

برقراری نیازمندی‌های امنیتی خواسته‌شده را به‌صورت خلاصه در زیر شرح می‌دهیم:

۱. پیام‌های رد و بدل شده بین کاربران همگی با استفاده از پروتکل‌های توضیح داده‌شده به صورت انتها به انتها رمز می‌شوند.

۲. در زمان توافق کلید بین دو کاربر، به‌دلیل استفاده از نانس و برچسب زمانی ویژگی تازگی برقرار خواهد بود. از طرفی خودِ پروتکل Double Ratchet حافظ ویژگی تازگی برای ما خواهد بود زیرا برای هر پیام از یک رمز جدید استفاده می‌شود.

۳. هر پیامی که بین دو کاربر و بین کاربر و سرور ارسال می‌شود توسط کلید خصوصی رمز می‌شود تا صحت و یکپارچگی پیام‌ها حفظ شود.

۴. به دلیل استفاده از برچسب زمانی برای ارسال پیام‌ها سازگاری آن‌ها هنگام ارسال برقرار خواهد بود.

۵. برای ویژگی‌های احراز اصالت و عدم انکار تمامی پیام‌ها توسط کلید نامتقارن امضا می‌شوند و امضا به همراه پیام ارسال می‌شود.

۶. امکان اضافه‌کردن عضو جدید به گروه تنها توسط ادمین‌ها قابل انجام است و در سمت سرور این درخواست فقط از کاربران ادمین پذیرفته می‌شود.

۷. حمله مرد میانی نیز به دلیل رمزگذاری انتها به انتها بین هر دو کاربر و همچنین رمزگذاری بین کلاینت و سرور امکان‌پذیر نخواهد بود.

۸. حمله تکرار نیز مشابها به دلیل استفاده از الگوریتم Double Ratchet و برچسب زمانی امکان‌پذیر نخواهد بود.

۹. الگوریتم‌های رمزگذاری استفاده شده همگی از روش‌های بروز و مورد تایید هستند و در حال حاضر امن می‌باشند.

۱۰. ویژگی محرمانگی پیش‌رو به دلیل استفاده از پروتکل Double Ratchet برقرار می‌باشد. زیرا در صورت لو رفتن کلید بلندمدت کلیدهای نشست قبلی قابل بازیابی نخواهند بود و محرمانگی پیام‌هایی که درگذشته تبادل شده‌اند حفظ می‌شود.

۱۱. محرمانگی پس‌رو نیز به‌دلیل استفاده از پروتکل Double Ratchet برقرار خواهد بود.