پیاده سازی های لایه سوکت امن - یک بررسی

چکیده- یک لایه سوکت امن (SSL)، یک پروتکل است که توسط Netscape برای انتقال اسناد خصوصی به صورت ایمن بر روی اینترنت توسعه یافته است. SSL را می توان به طور موثر برای محافظت از داده ها در انتقال مورد استفاده قرار داد. پروتکل SSL در بین پروتکل لایه کاربرد (به عنوان مثال، HTTPS ها (انتقال بیش از حد متن پروتکل امن)) و پروتکل لایه حمل و نقل می آید. واسط های نرم افزار HTTP با SSL تقریبا در همان روشی قرار می گیرد که با TCP در غیاب امنیت است. تا آنجا به TCP مربوط می شود، SSL فقط یک پروتکل نرم افزار با استفاده از خدمات آن است. SSL از دو زیر پروتکل، پروتکل دست دهی و پروتکل رکورد تشکیل شده است. قدرت SSL و از این رو عملکرد ارائه شده توسط ارتباط امن با انتخاب مجموعه رمز تعیین می شود. مجموعه رمز به خودی خود به چهار اجزا رسیده است و آنها روشی برای تبادل کلید، احراز هویت، رمزنگاری و روشی برای محاسبه مخلوط هضم پیام هستند. روش های مختلف برای همه آنچه در روش بالا گفته شده است در دسترس است و ما نیاز به انتخاب یک مجموعه رمزنگاری داریم که کاملا با عملکرد ، محدودیت سرعت و حافظه مورد نیاز ما مطابقت خواهد کرد. موارد ساخته شده مختلف در کتابخانه ها برای توسعه دهندگان به منظور استفاده وجود دارد. این مقاله مقایسه و شرح مختصری در مورد پیاده سازی SSL معمولا بیشتر استفاده شده مانند OpenSSL ها، CyaSSL و MatrixSSL را فراهم می کند.

کلمات کلیدی: لایه سوکت های امن (SSL) امنیت لایه انتقال (TLS)؛ امنیت لایه حمل و نقل داده ای؛ پروتکل دست دهی، پروتکل رکورد؛ استاندارد رمزگذاری پیشرفته (AES).

I. مقدمه

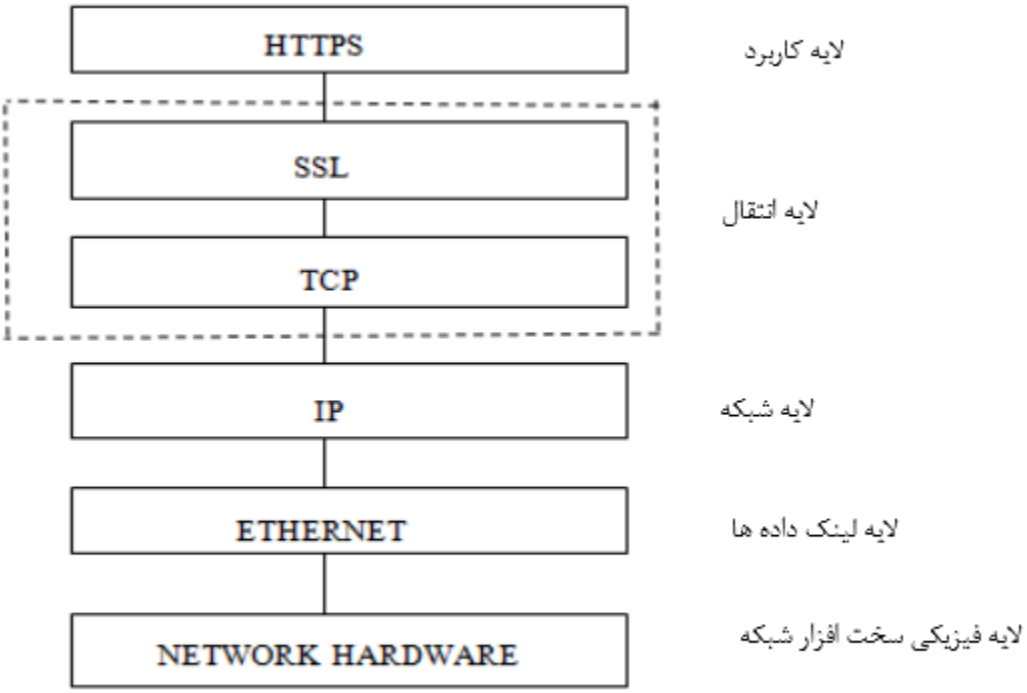
اینترنت از TCP / IP برای انتقال داده ها استفاده می کند و پروتکل TCP / IP یک مکانیزم امنیتی / حفاظت ذاتی ندارد. بنابراین داده ها در انتقال ممکن است توسط حملات فعال مانند جعل پیام و تغییر پیام [1] تحت تاثیر قرار گیرند. به منظور حفاظت از داده ها در انتقال، Netscape مفهوم پروتکل SSL را در سال 1994 معرفی کرد و بعد از آن شرکت های دیگر مانند مایکروسافت شروع به توسعه پروتکل های امنیتی خود نمودند. سپس نیروی ضربت مهندسی اینترنت (IETF) در تعریف یک استاندارد برای یک پروتکل رمزنگاری-لایه مداخله نمود. با ورودی از فروشندگان متعدد، IETF لایه حمل و نقل استاندارد را ایجاد نمود. نسخه های قبلی از SSL 2.0 SSL و: SSL 3.0.Transport لایه امنیتی (TLS) پروتکل مبتنی بر SSL 3.0 و TLS 1.0 همان SSL TLS 3.1.Eventhough است و پروتکل SSL کمی در اجرای آنها متفاوت است، توسعه دهنده نرم افزار و کاربر نمی تواند هر گونه تفاوت را [2] تشخیص دهد.

هر شبکه ای است که نیاز به انتقال داده ها بر روی یک شبکه نا امن مانند اینترنت دارد می تواند از این پروتکل SSL برای اطمینان از امنیت استفاده نماید. امروزه SSL به طور گسترده ای در معاملات بانکداری آنلاین (که در آن ما نیاز به حفاظت از اعتبار کاربر ارزشمند مانند رمز عبور، شماره PIN و غیره داریم)، مقاصد نظامی، سیستم های جاسازی شده و غیره استفاده می شود که ارتباطات امن بین کلاینت و سرور را فراهم می کند. در اینجا، مشتری مرورگر و سرور ما، سرور وب است که با ما در حال برقراری ارتباط [3] است. یک ارتباط در صورتی امن نامیده می شود که محرمانه بودن، احراز هویت و تمامیت پیام را تضمین نماید [4]. محرمانه مورد استفاده قرار گیرد تا اطمینان حاصل شود که تنها افراد مجاز در حال خواندن پیام هستند. توسط ویژگی یکپارچگی، این معنی را دارد که پیام در انتقال تغییر نمی کند. احراز هویت با استفاده از گواهینامه های دیجیتال به دست می آید و از آن استفاده شده است تا اطمینان حاصل شود که ما در حال برقراری ارتباط با کاربران واقعی هستیم و نه هر مزاحمی. در این مقاله شرح مفصلی از پروتکل SSL داده شده است. شروع با ساختار SSL، جایگاه آن در معماری شبکه، تهدیدات

امنیتی در SSL و بخش بعدی از این مقاله، پیاده سازی مختلف های SSL مانند MatrixSSL، OpenSSL، و CysSSL و مطالعه تطبیقی آنها را پوشش می دهد.

II. ساختار SSL

SSL در بین لایه کاربرد HTTP و TCP می آید. شکل 1 موقعیت SSL در معماری شبکه را نشان می دهد. SSL نیاز به چند تغییر برای پروتکل های بالا و پایین آن دارد [5]. واسطه های نرم افزار HTTP با SSL تقریبا در همان روشی عمل می کنند که TCP در فقدان امنیت عمل می کنند. تا آنجا به عنوان TCP مربوط می شود، SSL فقط یکی دیگر از نرم افزار است که از خدمات آن استفاده می کند. SSL از دو زیر پروتکل تشکیل شده است، پروتکل دست دهی و پروتکل رکورد.



شکل ۱. موقعیت SSL در معماری شبکه

A. پروتکل دست دهی

دلیل نام نهادن پروتکل دست دهی اینست که عملیات دست دهی اولیه از قبیل گواهینامه های تبادل، کلید تبادل مواد، و تصدیق هویت را انجام می دهد. گام های مختلف درگیر در پروتکل دست دهی به شرح زیر [6] ارائه می شوند،

• مشتری یک پیام ClientHello را به یک سرور برای درخواست اتصال به SSL می فرستد. این پیام ClientHello بالاترین نسخه از SSL که از آن پشتیبانی می کنند، یک عدد تصادفی، یک لیست از درخواست های رمزنگاری پیشنهادی و یک روش فشرده سازی که مشتری می تواند از ان حمایت کند را مشخص می کند.

• سرور با یک پیام ServerHello، حاوی نسخه انتخاب پروتکل، یک عدد تصادفی، مجموعه رمزنگاری، و استفاده از روش فشرده سازی از گزینه های ارائه شده توسط مشتری پاسخ می دهد. سرور نیز ممکن است یک ID جلسه را به عنوان بخشی از این پیام برای انجام دست دهی ارسال کند.

• سرور پیام گواهی آن را (بسته به مجموعه رمز انتخاب شده، این ممکن است توسط سرور حذف شود) می فرستد.

• سرور یک پیام ServerHelloDone را می فرستد، که نشان می دهد که مذاکره دست دهی کامل شده است.

• پاسخ مشتری با یک پیام ClientKeyExchange است، که ممکن است شامل یک راز پیش اصلی، کلید عمومی، و یا هیچ چیزی باشد. (این مورد به رمز انتخاب شده بستگی دارد.)

• مشتری و سرور از اعداد تصادفی و راز پیش اصلی برای محاسبه یک راز مشترک، به نام "راز اصلی" استفاده می کند، و سپس هر دو طرف با استفاده از راز اصلی برای محاسبه یک کلید بلوک استفاده می کنند. تمام اطلاعات کلیدی دیگر برای این اتصال از این کلید بلوک (کلاینت و سرور تولید اعداد تصادفی) است، که از طریق یک تابع با دقت طراحی شده " شبه تصادفی" منتقل شده است. اطلاعات کلیدی شامل دو کلید های جلسه می شود:

client_write_key و server_write_key

• مشتری در حال حاضر یک رکورد ChangeCipherSpec، اساسا برای گفتن به سرور، "همه چیزی که من به شما می گویم از حالا به بعد رمزگذاری خواهد شد" را می فرستد." ChangeCipherSpec به خودی خود یک پروتکل در سطح رکورد است.

• در نهایت، مشتری یک پیام نهایی را که حاوی یک رشته hash و MACK روی پیام های دست دهی قبلی است تولید می کند و کدگذاری با استفاده از client_write_key قبل از ارسال آن صورت می گیرد.

• سرور برای رمزگشایی پیام پایان یافته مشتری توسط client_write_key تلاش می کند و hash و MACK را بررسی می کند. اگر رمزگشایی و یا تایید خراب شود، دست دهی شکست خورده در نظر گرفته می شود و این اتصال باید پاره شود.

• در نهایت، سرور یک ChangeCipherSpec ، پیام رمزگذاری شده آن را می فرستد و مشتری رمزگشایی و تایید را انجام می دهد. در این نقطه، "دست دهی" کامل است و پروتکل نرم افزار فعال شده است. برنامه پیام های رد و بدل شده بین کلاینت و سرور رمزگذاری خواهد شد.

B. پروتکل رکورد

در پروتکل رکورد از کلید جلسه تولید شده در پروتکل دست دهی برای محفظه ای نمودن داده هایی که باید رد و بدل شود استفاده می شود. بسته بندی نمودن برای داده ها می تواند محرمانه بودن و تمامیت داده ها را ارائه کند.

C. درخواست های رمز

قبل از ارسال داده ها، فرستنده و گیرنده باید به یک نتیجه گیری در مورد اجزای مجموعه رمز انتخاب شده برسد [7]. چهار جزء از مجموعه رمزنگاری، الگوریتم تبادل کلید هستند؛ روش احراز هویت، روش های رمزنگاری و الگوریتم برای محاسبه پیام هضم هش. الگوریتم تبادل کلیدی مشخص می کند که چگونه همتایان بر یک کلید متقارن مشترک که می تواند برای به رمز در آوردن پیام پس از handshaking مورد استفاده قرار گیرد به توافق می رسند. دو الگوریتم تبادل کلید مشترک DHE (الگوریتم Diffie-HellmanKeyExchange) و RSA (Rivest، Shamir-Adelman). الگوریتم احراز هویت نشان می دهد که چگونه مشتری و سرور، هویت های خود را به یکدیگر ثابت می کنند. گزینه های احراز هویت عبارتند از: RSA (Rivest، Shamir و Adelman)، DSA

(الگوریتم امضای دیجیتال)، منحنی بیضوی DSA، کلید از قبل مشترک (PSH) و ناشناس (زمانی که هیچ مکانیزم احراز هویت استفاده می شود). s\s، جزء رمزگذاری نشان می دهد که کدام الگوریتم متقارن می تواند برای به رمز در آوردن داده ها قبل از انتقال مورد استفاده قرار گیرد. RC4. (4 (RivestCipher، DES (استاندارد رمزگذاری داده ها)، DES (DES3) سه گانه)، AES (استاندارد رمزگذاری پیشرفته) هستند که معمولا استفاده می شوند. الگوریتم RC4 سریعترین و کوچکترین الگوریتم رمزنگاری است، و بنابراین برای پردازنده های جاسازی شده ایده آل است. و در نهایت اجزای مخلوط هضم مورد استفاده قرار می گیرند تا اطمینان حاصل شود که گیرنده آنچه را فرستنده منتقل نموده دریافت می کند، یعنی پیام در انتقال تغییر نمی یابد. SHA-1، الگوریتم کنترلی اغلب مورد استفاده برای تایید یکپارچگی داده های رد و بدل شده است.

III. تهدید به SSL پروتکل

پروتکل SSL مستعد در ابتلا به برخی از حملات یا تهدیدات امنیتی است. حمله عقبگرد ، نسخه های عقبگرد حمله، تبادل الگوریتم عقبگرد حمله و رها کردن تغییر تنظیمات رمز حمله محبوب تر هستند. بخش بعدی، شرح مختصری از چنین حملات ها و اقدامات ضد احتمالی است.

A. مجموعه رمز حمله عقبگرد

این نوع حمله اغلب در پروتکل SSL 2.0 ارز اتفاق می افتد که در آن مهاجم با پیام سلام مشتری برخورد می کند و لیست سوئیت رمزنگاری را که در یک متن ذخیره می شود تغییر می دهد. هنگامی که سوئیت های رمز تغییر یابند، در حال حاضر مهاجم می تواند در واقع کاربر را وادار به استفاده از رمزگذاری صادر شده ضعیف نماید، حتی اگر هر دوی سرور و مشتریان الگوریتم درجه قوی تر را ترجیح دهند و از آن پشتیبانی نمایند. ما می توانیم این مشکل را با استفاده از نسخه بالاتر از SSL (SSL 3.0) و از جمله مقادیر هش برای همه پیام ها در طول پروتکل

SSL دست دهی و تصدیق آن را در پیام به پایان رسیده با تایید ارزش هش و پیام کاهش دهیم. قرار نیست هر دو سرور و مشتری داده های کاربردی را تا زمانی که پیام به پایان رسیده از هر دو طرف تایید شود قبول نمایند [7].

B. حمله عقبگرد نسخه

حمله عقبگرد نسخه، حمله ای است که از یک مکانیسم استفاده می کندکه باعث می شود نسخه بالاتر مشتریان (V 3.0) و سرور به سقوط به نسخه پایین تر (V 2.0) مستعد شود. سرور فریب خورده فکر می کند که او در حال برقراری ارتباط با یک مشتری است که فقط از SSL 2.0 پشتیبانی می کند. دلیل آن این است که SSL 2.0 دارای مغالطه و نقص های خود است. نسخه SSL 2.0 از پروتکل شامل پیام های "به پایان رسیده" نمی شود که باعث می شود حمله گر از آسیب پذیری های SSL 2.0 بهره برداری نماید. این مشکل می تواند توسط مشتری برای پیاده سازی ثابت افزونگی بایت های RSA PKCS شناسایی شود که از SSL 3.0 پشتیبانی می کند. سرور که از SSL 3.0 پشتیبانی می کند مخالف پذیرش تبادل کلید رمزگذاری شده RSA روی SSL 2.0 است تنها در صورتی آن کلمه ها در ادامه متن در رمزنگاری RSA گنجانده شده باشند [7].

C. تبادل الگوریتم عقبگرد

در این نوع حمله، مهاجم پیام سلام مشتری را در صورتی ره گیری می کند که اگر متن باشد و سعی خواهد کرد که آن با یک الگوریتم تبادل کلید ضعیف و یا به طور تصادفی انتخاب شده جایگزین نماید. مهاجم همان را در پیام از یک سرور به یک مشتری تکرار می کند و با یک الگوریتم که او می خواهد جایگزین می کند. [8] ما می توانیم این نوع حمله را با انتقال پارامترهای امضا شده بهبود دهیم به طوری که مهاجم هرگز نمی تواند یک الگوریتم را جایگزین نماید.

D. حذف تغییر رمز تنظیمات حمله

پیام تغییر رمز تنظیمات توسط سرور و کلاینت برای اطلاع یکدیگر فرستاده می شود که پیام های بعدی توسط تنظیمات رمزنگاری و کلیدهای محافظت به تازگی مذاکره شده فرستاده خواهد شد. این یک واقعیت است که تنظیمات رمزنگاری از محاسبه MAC از پیام های قبلی دست دهی حذف می شود که به عنوان یک کد تأیید هویت پیام نهایی استفاده می شود. مهاجم در واقع می تواند پیام تنظیمات تغییر رمزگذاری را حذف نماید که باعث می شود کاربر و سرور هرگز از خواندن حالت خواندن در انتظار به وضعیت فعلی تغییر نکند و تأیید هویت پیام و رمزگذاری در لایه ضبط غیرفعال شود. ما می توانیم این مشکل را با چک کردن تنظیمات پیام تغییر رمز قبل از این پیام به پایان برسد تایید شود و همچنین با استفاده از SSL version3.0 کاهش دهیم که برای اصلاح این نقص [9] طراحی شده است.

IV. کتابخانه های رایج SSL

محبوب ترین اجرا از TSL پروتکل (حمل و نقل امنیت لایه) CyaSSL، OpenSSL و MatrixSSL می باشند. MatrixSSL به خودی خود در دو نسخه آزاد و تجاری می آید. شرح مختصری از هر یک در زیر آورده شده است.

A. OpenSSL

OpenSSL بر اساس کتابخانه SSLeay که بسیار عالی توسط Eric A. Young و Tim J. Hudon توسعه یافته است می باشد. ویژگی اصلی پروژه OpenSSL اینست که دارای کامل امکانات و ابزار منبع باز برای اجرای لایه سوکت پروتکل ایمن Secure Socket Layer (SSL v2/v3) و امنیت لایه حمل و نقل (TLS V1) پروتکل های [10] می باشد. آخرین نسخه از OpenSSL OpenSSL c1.0.1 تحت مجوز Apache-style است، و از این رو توسعه دهندگان به طور رایگان آن را دریافت و منوط به برخی شرایط مجوز ساده از آن برای مقاصد تجاری و غیر تجاری استفاده می کنند. ابزار OpenSSL را می توان برای اهداف زیر استفاده نمود:

- ایجاد و مدیریت پارامترها و کلید خصوصی، کلید عمومی.
- عملیات عمومی رمزنگاری کلیدی.

• ایجاد گواهی X.509، CSRs ها و CRLs.

• محاسبه Message Digests.

• رمزگذاری و رمزگشایی با رمزها.

• آزمون های SSL / TLS کاربر و سرور.

• هدایت S / MIME (فرمت ایمیل اینترنت چند منظوره) امضا و یا ایمیل های رمزگذاری شده.

• درخواست تمبرهای زمانی، تولید و تایید

OpenSSL از تعدادی از الگوریتم های مختلف رمزنگاری: رمزهای های پشتیبانی شده توسط OpenSSL AES (استاندارد رمزگذاری پیشرفته)، Blowfish، Camellia، SEED (این رمز بلوک توسط آژانس امنیت کره ای توسعه یافته)، CAST-128 (متناوب به عنوان CAST5 شناخته می شود، که یک بلوک رمزی است که نام خود را از مخترعین کارلایل آدامز و استافورد تاوارس گرفته است)، DES (استاندارد رمزگذاری داده ها)، IDEA (الگوریتم بین المللی رمزگذاری داده ها)، RC2 (Rivest رمز 2) RC4، RC5، سه گانه DES و GOST 28147-89. زیر توابع هش رمزنگاری را می توان با OpenSSL MD5 (پیام Digest5)، MD2، SHA-1 استفاده شده (، SHA-2، RIPEMD-160، MDC-2، GOST R 34.11-94 پشتیبانی می کند.

B. CyaSSL

CyaSSL یک کتابخانه SSL تعبیه شده است که سبک وزن است و در ANSI C [11] نوشته می شود. هدف آن، جاسازی و عمدتا محیط های سیستم عامل زمان واقعی (RTOS) است، به دلیل اندازه کوچک، سرعت آن، و مجموعه ای از ویژگی ها. CyaSSL از استانداردهای صنعت درسطح فعلی TLS 1.2 و DTLS (دادهای حمل و نقل امنیت لایه) پشتیبانی می کند. تا حدود 20 برابر کوچکتر از OpenSSL است و رمزهای مترقی مانند HC-128، RABBIT، و NTRU را ارائه می دهد. پشتیبانی کامل از مشتری و سرور را ارائه می دهد. پشتیبانی OCSP و CRL نیز در CyaSSL وجود دارد. اندازه آن در محدوده مورد نیاز حافظه 30 kB.100 است. الزامات

حافظه آن 3-kB.The36 است. اولین کاربر عمده yaSSL / CyaSSL، MySQL است، محبوب ترین پایگاه داده متن باز در جهان. CyaSSL در حال حاضر برای Win32/64، لینوکس، سیستم عامل مک ایکس، سولاریس، FreeBSD و NetBSD، عامل ها، لینوکس تعبیه شده، Haiku، WRT Open، آی فون، آندروید، رشته مهندسی نینتندو گیمکیوب و از طریق حمایت DevKitPro، QNX، VxWorks، Monta ویستا، ThreadX، انواع TRON، از OpenCL، و سیستم عامل Micrium'sMicroC / II FreeRTOS، فری اسکیل Freescale MQX، و Nucleus در دسترس است.

C. MatrixSSL

MatrixSSLیک SSL جاسازی شده و اجرای TLS طراحی شده برای برنامه ها و دستگاه های کاربردی کوچک ردپا است [12]. به عنوان یک منبع منبع باز در دسترس است و کاملا با الزامات سیستم های جاسازی شده (محدودیت حافظه، قابلیت پردازش و قدرت باتری محدودیت) مطابقت دارد. MatrixSSL به سیستم های عامل از جمله VxWorks، uClinux، eCos، FreeRTOS، ThreadX، ARM، MIPS32، Power، H-8، SH3، i386 و ویژگی های های .x86-64 پورت شده است. ویژگی های مهم MatrixSSL، در زیر آمده است:

• رد پای کلی MatrixSSL با فراهم کننده رمزنگاری کمتر از KB50 است.

سرورهای SSL 3.0 و TLS 1.0 و 1.1 و پشتیبانی مشتری.

• شامل سازمانهای سری ومخفی کتابخانه - RSA و ECC، 3DES، AES، ARC4، SHA1، MD5، RC2

• رمز سوئیت - RC4-MD5، SHA RC4، DES-CBC3-SHA، AES128-SHA، AES256-SHA.

• پشتیبانی کامل از سرگیری / ذخیره جلسه.

• کلید زنی دوباره جلسه و مذاکره دوباره رمز.

• سرور و مشتری X.509 گواهی تأیید اعتبار های زنجیره ای.

• تجزیه X.509 PEM و ASN.1 فرمت گواهی DER.

• پشتیبانی PKCS # 1.5، PKCS # 8 PKCS # 5 و PKCS # 12 برای قالب بندی کلیدی.

• پشتیبانی از خط فرمان SSH.

• پلت فرم کاملا متقابل، کدهای قابل حمل، حداقل استفاده از تماس های سیستم.

• رابط مجموعه رمزهای نردبانی.

• رابط ارائه دهنده سازمانهای سری و مخفی نردبانی.

• سیستم عامل های نردبانی و رابط malloc.

• TCP / IP اختیاری.

• زنجیره ای از دستورهای اختیاری.

• تنها یک رابط برنامه کاربردی خارجی API که همه آنها غیر مسدود است.

• کد تمیز به شدت اظهار نظر شده در قابل حمل C

V. مقایسه کتابخانه های SSL / TLS

کتابخانه SSL به طور معمول استفاده شده در الزامات حافظه، ویژگی های امنیتی ارائه شده، هزینه و مشکل در پیاده سازی و غیره به طور قابل توجهی متفاوت است. در این بخش یک مطالعه مقایسه ای از آنها ارائه شده است. "جدول 1" یک مرور کلی از سه کتابخانه ذکر شده در بالا را ارائه می دهد.

جدول I مرور کلی

پیاده سازی	توسعه یافته	منبع باز	لیساتس نرم افزار	آخرین نسخه پایدار	تاریخ واگذاری
CyaSSL	CyaSSL	Yes	GPLv2 &commercial license	2.4.0	10-10-12
OpenSSL	OpenSSL project	Yes	OpenSSL/SSLeay dual license	1.0.1c	10-05-12
MatrixSSL	PeerSec Networks	Yes	Proprietary	3.3	22-02-12
MatrixSSL-open	PeerSec Networks	Yes	GPLv2(General public License version 2)	3.3	22-10-12

A. پشتیبانی از پروتکل

چندین نسخه از پروتکل TLS مانند SSL 2.0، SSL 3.0 TLS1.0 وجود دارد، اما عملکرد ارائه شده توسط هر یک از آنها به طور قابل ملاحظه ای متفاوت است. به عنوان مثال SSL 2.0 یک پروتکل توصیه شده، در برابر حملات آسیب پذیر است. در مقایسه با SSL 2.0، SSL 3.0 و TLS1.0 عملکرد بهتری ارائه می دهد و هر گونه آسیب پذیری های عمده شناخته شده را ندارد. TLS 1.2 آخرین نسخه منتشر شده است که عرضه کننده ویژگی های جدید است. دادهای TLS (DTLS) یک اصلاح از TLS 1.1 برای یک لایه حمل و نقل بسته گرا است؛ در اینجا ما نیاز به در نظر گرفتن افت بسته ها و مرتب سازی مجدد بسته ها داریم. مقایسه کتابخانه های SSL بر اساس پروتکل هایی که آنها پشتیبانی می کنند، در "جدول شماره 2" داده شده است.

جدول II پشتیبانی از پروتکل

Implementation	SSL 2.0	SSL 3.0	TLS 1.0	TLS 1.1	TLS 1.2	DTLS1.0	DTLS 1.2
CyaSSL	No	Yes	Yes	Yes	Yes	Yes	No
OpenSSL	Yes	Yes	Yes	Yes	Yes	Yes	No
MatrixSSL-open	No	Yes	Yes	Yes	No	No	No
MatrixSSL	No	Yes	Yes	Yes	Yes	Yes	Yes

REFERENCES

[1] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[2] S.Thomas, "SSL and TLS essential", John Wiley & Sons, inc, 2000.

[3] W.B.Mao,"Modern Cryptography:Theory and Practice,"Publishing House of Electronics Indudtry,Beijing,2004

[4] Zhai Xuefeng, "The security analysis of SSL and the research and realization of its being hijacked," Sichuan University,2004.

[5] Qianqian Ge,Feng Chen,"Strategies for Implementing SSL on Embedded System",2008 International Seminar on Future BioMedical Information Engineering.

[6] G. Apostolopoulos, V. Peris V,D. Saha, "Transport Layer Security How much does it really cost," Infocom'99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings, IEEE,1999,pp. 717-725

[7] David Wagner, Bruce Schneier, "Analysis of the SSL 3.0 protocol", USENIX Workshop on ElectronicCommerce,ACM.

[8] Korea Information Security Agency, "A development of modules for Improving an ability and security of SSL/TLS", January, 2002

[9] www.jucs.org Embedded Monitors for Cryptographic Protocol Intrusion Detection and Prevention http://www.jucs.org/jucs_11_1/protomon_embedded_monitors_for/Joglekar_S_P.html.Retrieved:December5,2009.

[10] The Open Source Toolkit for SSL/TLS .www.openssl.org.

[11] Embedded SSL Library for Applications, Devices, and the Cloud, http://www.yassl.com/yaSSL/Home.html

[12] MatrixSSLSpecification,www.matrixssl.org