مفاهیم SSL ، امضای دیجیتالی
و مراکز صدور گواهینامه

SSL (Secure Sockets Layer), Digital Signature And CA (Certificate Authority) Fundamentals

محمود مروج
Mahmoud@Moravej.ir
دانشجوی مهندسی کامپیوتر ، گرایش نرم افزار

۱- تعریف

برای شروع ، با تعریفی کلی از SSL که در سایت webopedia آمده است مطلب را آغاز می کنیم :

" SSL پروتکل ای است که توسط شرکت Netscape و برای رد و بدل کردن سند های خصوصی از طریق اینترنت توسعه یافته است.SSL از یک کلید خصوصی برای به رمز در آوردن اطلاعاتی که بر روی یک ارتباط SSL منتقل می شوند استفاده می نماید. هر دو مرورگر Internet Explorer و Netscape Navigator [و امروزه تمام مرورگر های مدرن] از این پروتکل پشتیبانی می نمایند. همچنین بسیاری از وب سایت ها برای فراهم کردن بستری مناسب جهت حفظ کردن اطلاعات محرمانه کاربران (مانند شماره کارت اعتباری) از این پروتکل استفاده می نمایند. طبق آنچه در استاندارد آمده است ، URL هایی که نیاز به یک ارتباط از نوع SSL دارند با https: به جای http: شروع می شوند.

پروتکل دیگری که برای انتقال اطلاعات به صورت امن بر روی شبکه جهانی وب وجود دارد ، پروتکل ای است به نام Secure HTTP یا S-HTTP . در حالیکه SSL یک ارتباط امن میان Client و Server ایجاد می کند تا هر اطلاعاتی که بر روی آن منتقل می شود امن باشد ، S-HTTP برای این طراحی شده است تا طبق آن پیام های

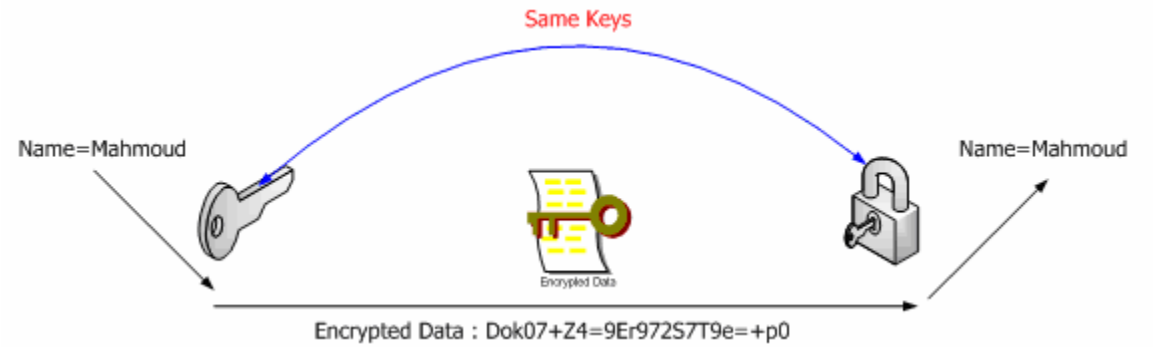
منفرد[۱] به طور امن منتقل شوند.بنابراین این دو تکنولوژی قبل از آنکه دو تکنولوژی رقیب باشند ، دو تکنولوژی مکمل هستند. هر دو ی این پروتکل ها به عنوان استاندارد از سوی IETF پذیرفته شده اند."

توجه داشته باشید که SSL یک پروتکل مستقل از لایه برنامه است (Application Independent) . بنابراین پروتکل هایی مانند HTTP ، FTP و Telnet قابلیت استفاده از آن را دارند. با این وجود SSL بروی پروتکل های HTTP ، FTP و IPSec بهینه شده است.

۲- مفاهیم رمز نگاری متقارن و نا متقارن

اساس رمز گذاری ها وجود کلید ها می باشند.بدین معنی که شما اطلاعات مورد نظر خود را توسط کلید قفل می کنید و سپس برای رمزگشایی آن مجددا از کلید استفاده می کنید. در رمز گشایی با کلید متقارن ، هر دو کلیدی که برای قفل و باز کردن اطلاعات استفاده می شود یکسان می باشد. بدین معنی که هر دو طرف از یک کلید یکسان بهره می برند که باید نزد خودشان امن باشد.
توجه کنید که مفهوم کلید در مباحث مرتبط ، عموما یک آرایه از بایت ها می باشد که بر اساس نوع امنیت طول متفاوتی دارد.مثلا ۰۱۱۰۱۱۰۰۱۱۰۰۱۰۱۱۱۰۰۱ می تواند یک کلید باشد.البته عموما کلید ها در مبنای ۱۶ نمایش داده می شوند. به هر حال وظیفه محافظت از کلید بر عهده دارنده آن است!

در شکل زیر نحوه رمز گذاری اطلاعات توسط کلید متقارن نمایش داده شده است:



اما نوعی دیگر از رمزگذاری وجود دارد که اساس SSL نیز می باشد.در این رمز گذاری که رمز گذاری نا متقارن یا رمز گذاری کلید عمومی نامیده می شود ، دو نوع کلید وجود دارد :

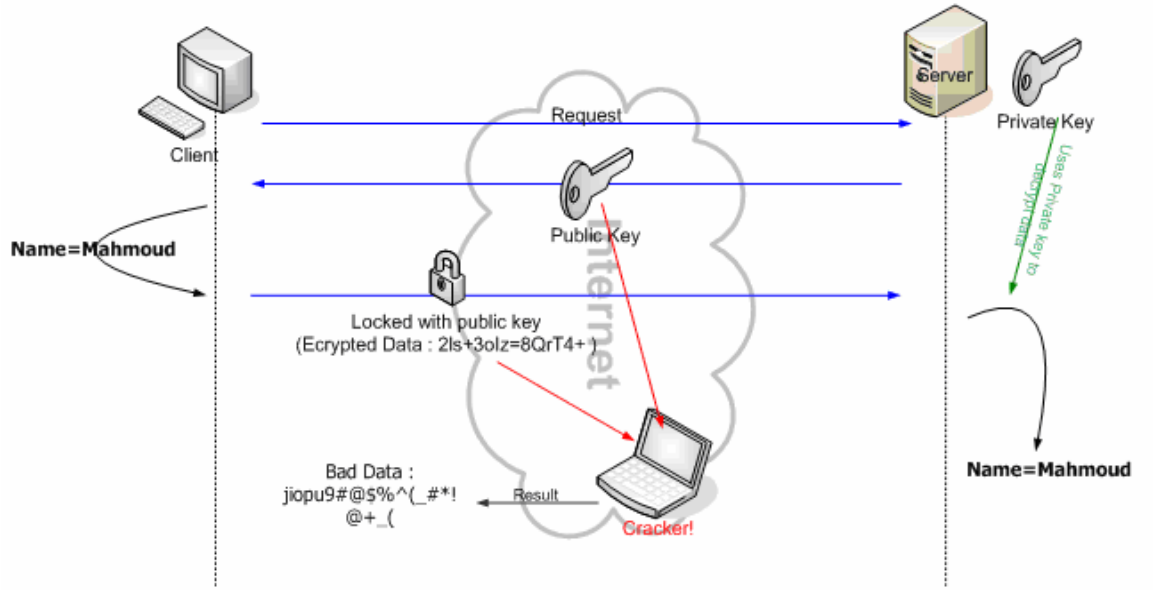
- کلید عمومی[۶]
- کلید خصوصی[۱۵]

در این رمز گذاری گفته می شود که اگر داده ای با یک کلید قفل شد ، با همان کلید باز نمی شود و فقط امکان باز شدن آن با کلید متناظر آن وجود دارد. این کلید متناظر نزد طرف مقابل است و امکان بدست آوردن آن از کلید دیگر وجود ندارد. به عبارت ساده تر اگر شما در خانه تان را با کلید A قفل نمودید ، تنها امکان باز شدن آن با کلید متناظر B وجود

دارد و این در حالیست که امکان فهمیدن آنکه کلید B چگونه ساخته شده است برای شما نیز وجود ندارد. حال اگر کلید خود را درون در نیز جا بگذارید ، مساله ای نیست!

حال به بحث باز می گردیم : شما درخواست داده ای را از یک سرور امن می کنید ، سرور کلید عمومی را برای شما ارسال می کند. شما داده های خود را با این کلید قفل می کنید و برای سرور ارسال می کنید. حال اگر این وسط کسی خواست داده ها را ببیند[۱۸] ، نمی تواند ، چراکه این داده ها با کلید عمومی باز نمی شوند!. در طرف مقابل سرور با کلید خصوص خود داده ها را رمز گشایی می کند و از آن استفاده می کند.

شکل زیر روند ذکر شده را می رساند :



توجه : در امضای دیجیتالی روند برعکس است.(به عبارت دیگر امضای دیجیتالی چیزی جز رمز گذاری داده ها با کلید خصوصی فرستنده نیست). ما در امضای دیجیتالی می خواهیم ببینیم که آیا داده های ارسال شده واقعاً از طرف شخصی است که ادعا می کند یا خیر؟

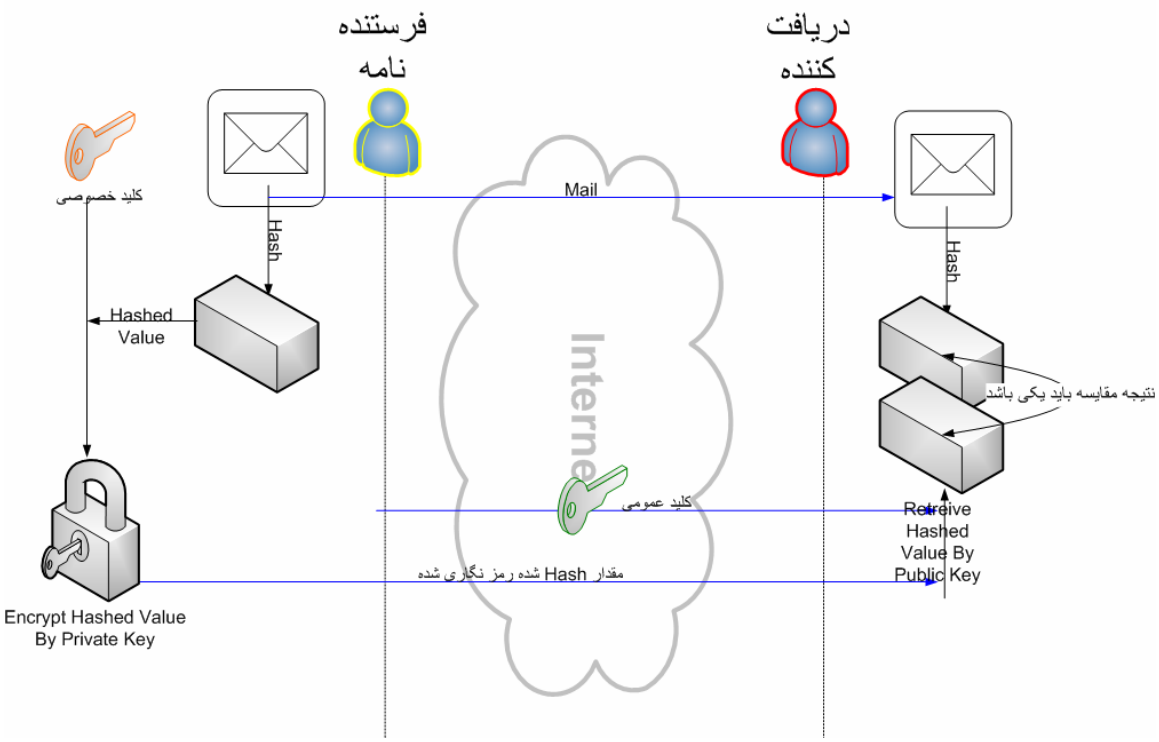
به طور ساده کاربر نام خود را با کلید خصوصی خود رمز گذاری می کند. در این حالت همه با کلید عمومی وی می توانند نام وی را رمز گشایی کنند و این صحیح است! چراکه هیچ کس دیگر قادر نیست داده ای تولید کند که نتیجه باز شدن آن با کلید عمومی شخص امضا کننده برابر باشد!.

البته در عمل بهتر است از توابع Hash استفاده می شود. چراکه در حالت فوق ، اولا نام کاربر را باید فقط کاربر و سرور بدانند و دیگر آنکه از کجا معلوم که داده ارسالی همانی است که امضای دیجیتالی کاربر با آن بوده است(به عبارت دیگر شاید در میان راه متن اطلاعات تغییر کرد)

همانطور که ذکر شد ، در این مورد از توابع Hash که یک طرفه هستند استفاده می شود.بدین معنی که اگر داده ای hash شد ، دیگر به هیچ عنوان (و توسط هیچ کلیدی) قابل برگشت نیست.

بدین منظور متن نامه ابتدا hash می گردد و سپس توسط کلید خصوصی امضا می شود.

سپس امضا و متن نامه ارسال می شود. در طرف سرور هم با کلید عمومی داده hash شده بدست می آید. متن ارسالی هم hash می شود. حال اگر این دو نتیجه یکسان بود ، داده ها واقعاً از طرف کسی که مدعی آن است ارسال شده است. چرا که اگر متن نامه عوض شود ، نتیجه hash آن هم متفاوت و مقایسه نتیجه یکسانی را بر نمی گرداند.
در نمودار زیر ، روند کار را مشاهده می کنید (قطعه های خاکستری رنگ نشان دهنده Hash شدن متن نامه می باشند)



۳ – ساختار و روند آغازین پایه گذاری یک ارتباط امن

در این پروتکل قبل از آنکه اطلاعاتی مابین درخواست دهنده و سرور رد و بدل شود ، می بایست ابتدا سرور تصدیق گردد[۲] .

به طور کلی مرحله آغازین شروع ایجاد ارتباط امن [۳] از دو فاز تشکیل شده است : تصدیق هویت سرور و مرحله اختیاری تصدیق هویت مشتری. در فاز تصدیق هویت سرور ، سرور در جواب درخواست مشتری گواهینامه خود و فرمول رمز گذاری خود[۴] را برای مشتری ارسال می کند. سپس مشتری یک کلید اصلی[۵] که با کلید عمومی[۶] سرور رمز گذاری شده است را تولید می کند و سپس این کلید رمز گذاری شده را به سرور ارسال می کند. سرور کلید اصلی را بازیابی می کند و خودش را با فرستادن پیغامی به مشتری تصدیق می نماید. درخواست های بعدی با کلید هایی که از کلید اصلی مشتق شده اند رمز گذاری و تصدیق می شوند.
در فاز دوم که اختیاری بود ، سرور یک چالش[۷] را برای مشتری ایجاد می کند [ارسال می کند]. مشتری نیز خودش را برای سرور با ارسال امضای دیجیتالی و گواهینامه کلید عمومی خود [۸] نسبت به تصدیق خود اقدام می نماید.

الگوریتم های زیادی جهت پنهان سازی در SSL استفاده می شوند. در مرحله آغازین شروع ایجاد ارتباط امن از الگوریتم RSA public-key cryptosystem استفاده می شود. بعد از رد و بدل شدن کلید ها نیز الگوریتم های متفاوتی استفاده می شوند. از جمله : RC۲ ، RC۴ ، IDEA ، DES ، triple-DES و MD۵ .
گواهینامه های کلید عمومی هم از قوانین X.۵۰۹ پیروی می کنند.(ساختار درختی CA ها و امضای گواهینامه ها که در ادامه ذکر خواهد شد ، همگی بر اساس این استاندارد است)

۴- پروتکل های مشابه

TLS[۹] هم پروتکل ای است که بسیار مشابه SSL ۳،۰ می باشد.

همچنین پروتکل WTLS[۱۰] که مخصوص شبکه های بیسیم است و در WAP[۱۱] استفاده می گردد.

۵- مفهوم گواهینامه در پروتکل SSL

در اینجا نیاز است که یک بحث کلی در مورد گواهینامه[۱۲] مورد نیاز این پروتکل صورت گیرد.به طور عموم (غیر از بحث SSL) گواهینامه ها جنبه اعتبار سنجی دارند. بدین معنی که اگر شما در یک بحث خاص دارای گواهینامه باشید ، به شما اعتماد بیشتری می کنند.اما ممکن است گواهینامه نداشته باشید ولی کار خود را هم به نحو احسنت انجام دهید.به طور مثال شما قهرمان مسابقات فرمول ۱ جهان هستید ، اما در صورتی که گواهینامه نداشته باشید ، هرگز اجازه نخواهید داشت که در شهر تردد کنید!

در مورد SSL هم تقریبا بحث به همین گونه است با این تفاوت که ذات این پروتکل با توجه به بحث گواهینامه ها طراحی شده است بدین معنی که اگر دارای گواهینامه نباشید ، قادر نخواهید بود که یک پیاده سازی از این پروتکل را داشته باشید.شاید در عالم راندن اتومبیل بدین صورت تعبیر شود که در صورتی که شما دارای گواهینامه نباشید ، قادر به رانندگی هم نیستید! این تشابه از جهاتی صحیح و از جهاتی غلط است . شاید برداشت صحیح تر به این صورت باشد که اگرچه قادر نخواهید بود بدون گواهینامه رانندگی کنید ، اما قادر هستید که خود برای خود یک گواهینامه صادر کرده و سپس به رانندگی بپردازید!(هرچند این گواهینامه از نظر دیگران کاملا بی ارزش است!).

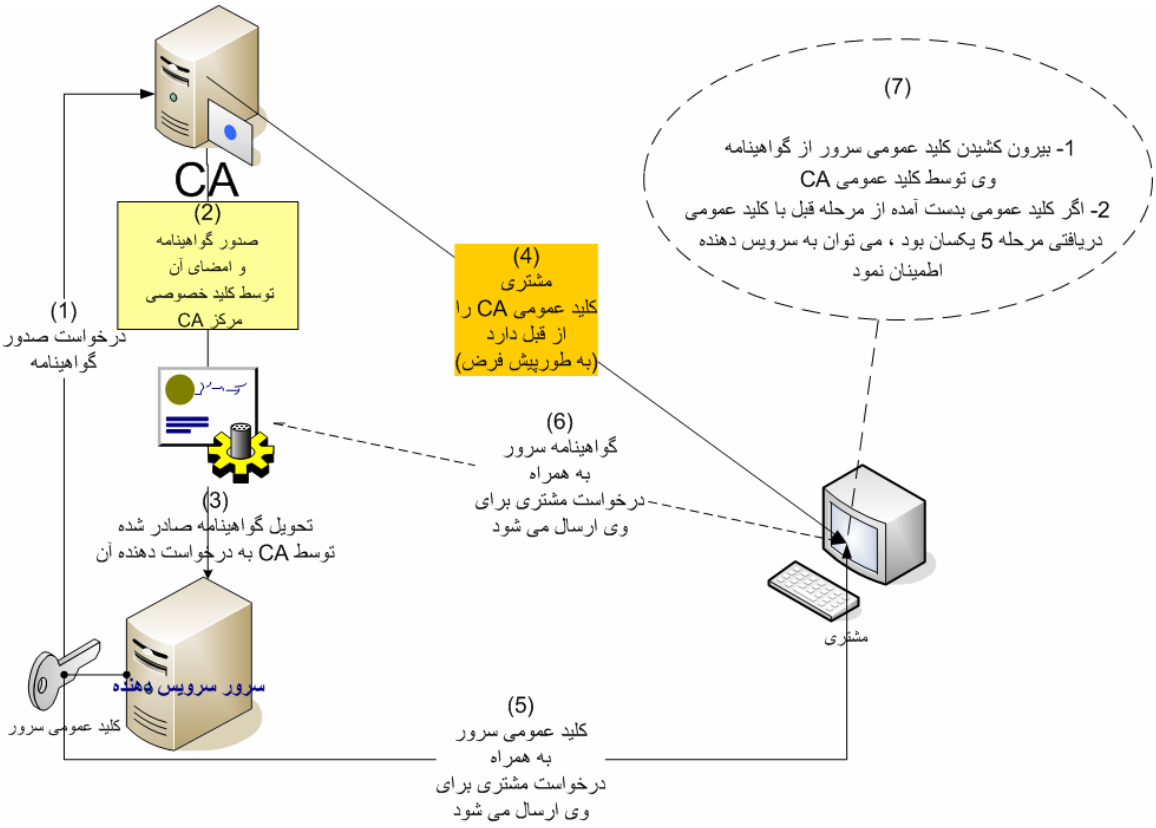
طبق بحث فوق ، شما قادر خواهید بود بدون پرداخت هیچ هزینه ای یک پروتکل SSL را راه اندازی و استفاده نمایید.نمونه بارز این استفاده در شبکه های داخلی یا Intranet می باشد.

۶- مراکز صدور گواهینامه

در SSL به مراکزی که اقدام به صدور گواهینامه می کنند ، "مرکز صدور گواهینامه "[۱۳] یا به اختصار CA گفته می شود.

این پروتکل از یک شخص ثالث [۱۴] (که همان CA می باشد) برای تشخیص هویت طرفین یک تراکنش استفاده می کند. در واقع یک گواهینامه معین می کند که آیا شخصی که دارنده آن است ، واقعاً همانی است که ادعا می کند یا خیر؟

در شکل زیر می توانید یک روند درخواست صدور گواهینامه توسط یک سرویس دهنده (قدم های ۱ ، ۲ و ۳) و در ادامه آن درخواست کاربر برای یک سرور دارای گواهینامه و چگونگی مطمئن شدن وی از معتبر بودن آن سرور را ببینید(قدم های ۴ ، ۵ ، ۶ و ۷) :



۷– مراحل کلی برقراری و ایجاد ارتباط امن در وب

به طور ساده مراحلی که در ایجاد یک ارتباط امن SSL در http طی می شود ، به صورت زیر می باشد :

۱- کاربر درخواست خود را از طریق مرورگر به یک صفحه امن ارسال می کند(آدرس این صفحه معمولا با https:// شروع می شود)

۲- وب سرور کلید عمومی[۶] خود را به همراه گواهینامه خود برای کاربر ارسال می کند.

۳- مرورگر چک می کند که آیا این گواهینامه توسط یک مرکز مورد اطمینان صادر شده است و اینکه آیا این گواهینامه هنوز اعتبار دارد ؟ و همچنین آیا این گواهینامه مرتبط با سایت درخواستی می باشد؟
۴- سپس مرورگر از این کلید عمومی [۶] دریافت شده از طرف سرور استفاده می کند سپس یک کلید متقارن[۱۴] تصادفی را تولید می کند و توسط آن تمام داده ها و URL را رمز گذاری می کند. در نهایت هم داده های رمز گذاری شده را به همراه خود کلید متقارن تولیدی ، مجددا توسط کلید عمومی سرور رمز گذاری کرده و نتیجه را به سرور ارسال می کند.
۵- وب سرور توسط کلید خصوصی[۱۵] خود، کلید متقارن رمز گذاری شده را رمزگشایی و با استفاده از آن سایر داده ها و URL را نیز رمزگشایی می نماید.
۶- وب سرور ، html درخواستی را با کمک کلید متقارن رمز گذاری و به کاربر باز می گرداند.
۷- مرورگر نیز داده های دریافتی را با کمک کلید متقارن خود بازگشایی کرده و به کاربر نمایش می دهد.

همانطور که از مرحله ۳ پیداست ، در این مرحله است که میزان اعتبار CA مشخص می شود. در صورتی که این CA به هر دلیل از نظر مرورگر دارای اعتبار و شرایط خاصی نباشند ، هشداری مبنی بر عدم امن بودن سایت مورد نظر به کاربر ارایه می دهد. توجه کنید که در این مورد تنها به هشدار بسنده می شود ، اطمینان به آن به شما و شرایط شما بستگی دارد. ضمن آنکه این هشدار هرگز نمی تواند به معنای قطعی عدم وجود امنیت باشد.حال اگر شما یک CA در اینترنت راه اندازی کردید ، مسلما هیچ کدام از مرورگر ها شما را نمی شناسند و بنابراین گواهی های صادر شده از طرف شما را نا امن می پندارند.از آنجا که کاربران عادی اینترنت نیز این هشدار ها را جدی در نظر می گیرند ، از ادامه تراکنش با سایت شما صرف نظر خواهند کرد.

۸– نکاتی در مورد گواهینامه ها

- شما در صورتی به یک سایت با یک گواهینامه معین اعتماد می کنید که آنرا یک CA معتبر (حداقل نزد شما) امضا کرده باشد.در واقع این اعتماد شما ضمنی است. به این روند ، درخت اعتبار گواهینامه[۱۶] یا مسیر گواهینامه[۱۷] گفته می شود. معمولا مرورگرها تعدادی از CA ها ی معروف را برای خود در نظر می گیرد.

- CA های متفاوتی در اینترنت وجود دارد که شاید مشهورترین آن verisign باشد.به هر حال قرار نیست شما همیشه ، با توجه به تراکنش خود، به تمام CA ها (یا به عبارت بهتر به انواع گواهینامه آنها) اعتماد کنید. یک راه مناسب برای تشخیص این موضوع میزان مبلغی است که گواهینامه مورد نظر تراکنش شما را بیمه می کند.به طور مثال حداکثر مبلغی که iranSSL تراکنش شما را بیمه می کند ۱۰،۰۰۰ دلار می باشد. اما Verisign گواهینامه ای دارد که تا ۲۵۰،۰۰۰ دلار تراکنش شما را بیمه می نماید.(بسیار مشابه با وضعیت شرکت های بیمه)

- پروتکل SSL بر اساس میزان امن بودن دسته بندی می شوند. این دسته بندی بر اساس مقدار bit های تولیدی به ازاء هر بخش از داده ای است که رمز گذاری می شود.مسلما هرچه تعداد این bit های تولیدی بیشتر باشد ، رمزگشایی آن بدون کلید ، بسیار سخت تر و با استفاده از کلید نیز زمان برتر خواهد بود. به

عنوان نمونه یک SSL با ۴۰ یا ۵۶ بیت (که یک رمز گذاری ضعیف می باشد) می تواند توسط یک هکر با ابزار کافی ، در عرض چند دقیقه شکسته شود. اما همین هکر برای مقابله با SSL ۱۲۸ بیتی ، نیاز به 2^{88} بار زمان بیشتر دارد! و این بدین معنی است که SSL ۱۲۸ بیتی نسبت به حالت ۴۰ یا ۵۶ بیتی ترلیون ترلیون بار امن تر و غیر قابل نفوذ تر است!

- یک بحث دیگر اینجا مطرح می شود و آن اینکه اگر یک هکر در میان راه کلید عمومی خود را جایگزین کلید عمومی سرور کرد. در این حالت عملا هکر به راحتی به اطلاعات کاربر دسترسی خواهد داشت. در واقع این دقیقا همان جایی است که لزوم وجود CA ها در پروتکل SSL مطرح می شود.در واقع CA ها کلید عمومی سرور را با کلید خصوصی خود امضا می کنند.مرورگر هم CA های قابل اعتماد را می شناسد(کلید عمومی آنها را دارد).این کلید عمومی سرور که توسط کلید خصوصی CA رمز گذاری شده است همان گواهینامه می باشد.از آنجا که سرور می بایست گواهینامه خود را ارسال کند ، در سمت مرورگر سعی می شود که توسط کلید های عمومی CA هایی را که می شناسد ، آن گواهینامه را رمز گشایی کند.اگر موفق شد و نتیجه با کلید عمومی سرور یکسان بود در واقع گواهینامه قابل اعتماد است.در این صورت امکان استفاده از گواهینامه دیگران هم وجود ندارد.(دقیقا همان بحث امضای دیجیتالی است)

- شرکت Verisign یک دوره آزمایشی مجانی برای کار با SSL می دهد که می توانید از طریق لینک زیر از آن بهره گیرید :

http://www.verisign.com/products-services/security-services/ssl/ssl-information-center/ssl-features-description/index.html

۹– واژه نامه

[۱] : Individual Messages
[۲] : Authenticate
[۳] : Handshaking
[۴] : Cipher Preferences
[۵] : Master Key
[۶] : Public Key
[۷] : Challenge
[۸] : Public-Key Certificate
[۹] : Transport Layer Security
[۱۰] : Wireless TLS
[۱۱] : Wireless Application Protocol
[۱۲] : Certificate
[۱۳] : Certificate Authority (CA)
[۱۴] : symmetric key

[۱۵] : private key

[۱۶] : certificate trust tree

[۱۷] : certificate path

[۱۸] : Sniffing

۱۰– فهرست منابع

۱- http://www.webopedia.com/TERM/S/SSL.html

۲- http://www.rsasecurity.com/rsalabs/node.asp?id=۲۲۹۳

۳- http://www.webopedia.com/TERM/S/S_HTTP.htm

۴- http://www.tldp.org/HOWTO/SSL-Certificates-HOWTO/x۶۴.html

۵- http://www.verisign.com/products-services/security-services/ssl/ssl-information-center/faq/index.html

۶- قدیر پور رستم ، مدل های اعتماد بر بستر کلید عمومی ، کتاب مقالات چهارمین همایش ملی دانشجویی انجمن کامپیوتر ایران ، ۱۳۸۱