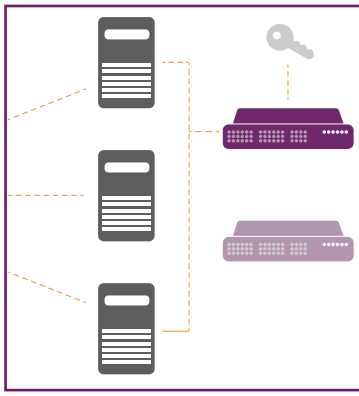
gemalto
security to be free



WHITEPAPER

Making SSL Faster and More Secure: Enhanced Hardware Key Management and Cryptographic Acceleration for SSL

The Secure Sockets Layer (SSL) protocol secures client-server communication sessions through the use of public key authentication and strong encryption. SSL's added security allows online transactions to be conducted over public networks, like the Internet, while maintaining the privacy of the data transmitted between the client and server. When used in conjunction with web-based applications, SSL allows transactions requiring the exchange of valuable or sensitive information, like banking, brokerage, and health care, to be conducted securely across the Internet.

SSL relies on the use of trusted digital credentials (in the form of an SSL private key and associated public key certificate) and both symmetric (also referred to as secret key) and asymmetric (also referred to as public key) cryptographic techniques to establish sessions between clients and servers. Unfortunately, SSL suffers from two potential weaknesses: first, the digital credentials used to authenticate the identity of a web server can be stolen or copied; second, the intensive cryptographic processes required to create SSL sessions can have a negative impact on web server performance.

To address these issues, new products have been developed—hardware security modules (HSMs) offer protection for digital credentials and SSL accelerators offload computationally-intensive cryptographic calculations, adding increased security and performance to applications relying on SSL. However, in many situations, having both the security of an HSM and the performance boost offered by an SSL accelerator is required. In these situations, a hybrid product offering both SSL acceleration and hardware key management is needed to meet the demands of applications that require absolute assurance for their SSL certificates while coping with large traffic loads.

SSL Overview

SSL is a security protocol that enables two computers to establish a secure, encrypted communication session to allow private information to be transmitted across open networks such as the Internet.

There are two primary issues that need to be dealt with when creating an SSL session: the first is establishing the identity of the people or computers at either end of the link; the second is the creation and exchange of keying material used to encrypt subsequent communications within the session. The methods used to accomplish these two operations rely on well-established cryptographic principles, and make use of proven encryption techniques to accomplish their goals.

Using SSL Credentials to Establish Identity

Participants in an SSL session use digital credentials to solve the problem of establishing identity. Digital credentials consist of a set of unique keys—a public key that is freely distributed on a digital certificate and a corresponding private key that is held only by the certificate owner. Certificates are issued by a central, trusted Certificate Authority (CA), either a private CA set up within a company to issue certificates to its employees, or a public or commercial CA that issues certificates to customers as a service. In either case, the CA scrutinizes an applicant's physical credentials, such as requiring proof of identity and a background check, before issuing a certificate. In the same way that physical identification, like passports, can be trusted due to the precautions in place to catch phony applicants, in the digital realm, certificates are used as identification based on the steps the CA has taken to verify the certificate owner's identity. Due to the trustworthiness and diligence of the issuing CA, anyone verifying a claimed identity using a digital certificate can trust that the identity is valid.

The SSL Handshake

When two computers negotiate a new SSL session, they use digital credentials to negotiate cryptographic protocols, authenticate themselves to each other, and exchange cryptographic material needed to create a unique session key used to encrypt subsequent communications.

- During the handshake process, the connecting client sends the server a list of acceptable cryptographic algorithms and a random number.
- Upon receipt of the client request, the server chooses the strongest algorithm contained in the client's list of supported cryptographic algorithms and sends back the chosen cipher suite, a second random number, and a copy of its SSL certificate (containing its public key).
- The client verifies the server's response by examining the contents of the certificate to ensure that it is valid, including checking the certificate's expiration date, matching the server's domain name against the name in the certificate, and verifying that the certificate issuer is a trusted Certificate Authority. Optionally, the client may send a copy of its own certificate to authenticate itself to the server.

After the client and server have agreed upon a set of cryptographic algorithms, and the client and server have successfully identified themselves with valid certificates, the client and server must still create and exchange a unique session key that will be used to encrypt their communications. The handshake protocol continues:

- To create the session key, the client generates another random number, called the premaster secret, which it encrypts with the server's public key (contained on the server's certificate) and sends it to the server.
- Once the server receives the pre-master secret, it decrypts it using its private key. Now, both the client and server have a copy of the pre-master secret that only they can know since the client created it and only the server can decrypt it. This implicitly authenticates the server to the client.

The handshake protocol is completed by using the random numbers exchanged earlier plus the pre-master secret to derive the master secret. The master secret provides the session key used to encrypt the SSL session. The server and client then exchange specially calculated message digests (known as an HMAC) of the complete contents of all handshake messages exchanged up to that point. Each side must verify the value of the HMAC received by the other before proceeding with the SSL-encrypted session. If the HMAC values do not match, the handshake process is restarted.

Missing Keys and Stolen Identities

At this point, a potential problem presents itself—what happens if an interloper gains access to the server's SSL private key? An intruder who has obtained a copy of a server's SSL private key can masquerade as that server and proceed with the rest of the handshake. A key thief could also establish a rogue website using the valid key holder's identity, and deceive unsuspecting victims lulled into a false sense of security by the identity presented to them. Even though all the checks and balances present within the SSL handshake process are performed, failure to secure the SSL private key brings doubt and uncertainty to SSL.

Because an SSL private key anchors the identity asserted by a corresponding SSL certificate, it is extremely important to ensure that this key remains secret and only accessible into its legitimate owner. In most cases, SSL private keys are stored as encrypted files on a computer's hard disk, where they are easily accessible to applications.

When the keys are needed, they are read from the files into the computer's memory, where they are used to perform cryptographic operations. While this method allows virtually anyone with a computer to establish a web server featuring SSL, the problem with this approach is that these encrypted files can be copied, manipulated, or deleted, providing an attacker with ready access when attempting to compromise a legitimate SSL server.

The deletion of a key amounts to a denial-of-service (DoS) attack—it creates inconvenience and downtime by depriving a business of SSL-based services, such as an e-commerce site's secure checkout or a bank's online banking services. In the case of the deletion of an SSL private key, its absence would be immediately evident and remedial measures could be quickly undertaken.

More insidious is an attack where the key is copied or manipulated; the attacker has the luxury of working undetected while causing more damage by using the key materials to masquerade as the legitimate site. These types of attacks are not trivial and require a high degree of technical proficiency on behalf of the attacker. As more business is conducted on the Internet, the incentive to exploit SSL security is increasing, particularly for organized crime and groups engaged in professional espionage with the motive and resources dedicated to illicit undertakings.

HSMs — A Proven Solution for Private Key Security

The problems surrounding SSL private key security for SSL certificates have already been experienced in the larger PKI space where the security of a single private key can impact the trust placed in thousands, if not millions, of certificates. Due to the high stakes surrounding the security of private keys used in PKIs, a new breed of specialized devices was developed called hardware security modules (HSMs). HSMs are designed to provide a more secure, hardware-based environment within which private keys are generated, stored, and used—eliminating the risks associated with storing private keys in a more vulnerable software repository. By providing physical and logical isolation of key materials from the computers and applications that use them, HSMs make it almost impossible to extract key materials through traditional network attacks. Additionally, tamper-resistant physical designs, coupled with strict operational policies, ensure that direct physical attack and attacks from trusted insiders are negated.

Given the burden of trust riding on private key security, strict validation and certification standards have been implemented by various government bodies to provide base criteria for the evaluation of HSMs. The most common standards are the National Institute of Standards and Technology (NIST) FIPS (Federal Information Processing Standard) 140-1/140-2 validation, and the multinational Common Criteria certification.

These standards provide a starting point for good HSM design by providing objective, third-party evaluation of the efficacy of an HSM's ability to protect private keys through stringent hardware, software, and operational design criteria. Fortunately, the solutions and technologies used to address private key security in PKI are transferable to the protection of SSL private keys that back SSL certificates. HSMs can be easily adapted to provide secure generation and storage for SSL private keys, preventing the compromise of keys by adding the assurance of hardware-secured key management to secure websites.

SSL Cryptography

SSL relies on several cryptographic processes to create secure, trusted links between a web server and its clients. The SSL certificate, and related SSL private key, are used during the authentication portion of the SSL handshake to prove the identity of the web server, and, optionally, the identity of the client, to ensure that the connecting parties are who they claim to be. Once this authentication is successfully completed, the client and server create and exchange a unique session encryption key. Finally, the session key is used to encrypt communications between the client and server.

Contrary to common assumptions, it is not the ongoing encryption of session data that taxes web servers offering SSL sessions, but rather a few operations performed during the SSL handshake. During the SSL handshake, both symmetric and asymmetric cryptographic techniques are used. Each technique has its own benefits and drawbacks, making each suitable to solve certain problems but not others.

Symmetric cryptography uses the same key to perform both encrypt/decrypt or sign/verify operations on messages. Symmetric cryptography is used during the SSL handshake to compute the message digests that are necessary to derive the master secret and to perform the final verification of the handshake protocol messages. Symmetric cryptography has the benefit of using smaller keys (for example, 128 bit) that make it relatively easy for computers to process; therefore, it is ideal in situations where a large amount of data must be encrypted/decrypted quickly. However, symmetric encryption has a large drawback resulting from the necessity for both parties to have a copy of the key, as evidenced by the lengthy handshake process required to securely create and exchange the SSL session key (in fact, asymmetric cryptography is needed to accomplish this exchange process securely).

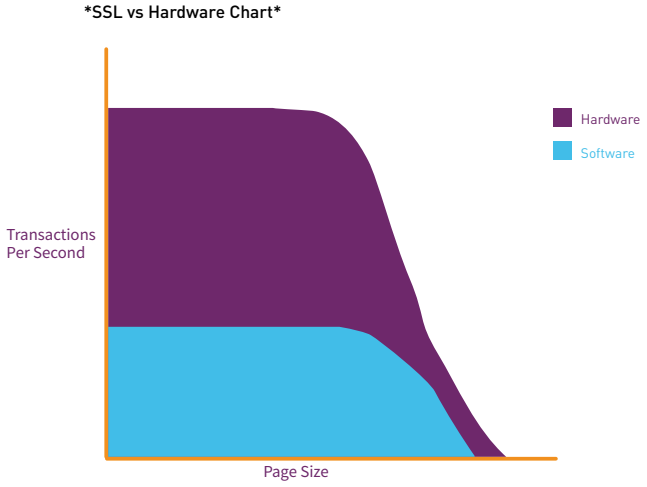
The other technique used by SSL is asymmetric cryptography, which uses a pair of keys—a public key to encrypt/verify messages and a separate private key to decrypt/sign them. Because the public key is only used to encrypt or verify documents, it can be freely distributed without concern, eliminating the elaborate security precautions required to exchange or agree on a secret key as used in symmetric cryptography. The drawback with asymmetric cryptography is that the private key must remain secret to only the key's owner. Additionally, the larger key sizes (for example, 2048 bit) used by asymmetric cryptographic operations increase the processing requirements for the asymmetric cryptography. Because of this processing penalty, asymmetric cryptography is only used during the SSL handshake. If all SSL communications employed asymmetric encryption, SSL sessions would slow to a crawl as both client and server would be over-burdened by cryptographic calculations.

Limits of Software-based SSL

Because the asymmetric cryptographic processing required during the SSL handshake is computationally intensive, web servers that use SSL to protect sessions can quickly become strained. Of special importance is the exchange of the pre-master secret from client to server since it relies on asymmetric operations. During the pre-master secret exchange, the client encrypts the pre-master secret with the server's asymmetric SSL public key. Conversely, the server must decrypt the encrypted pre-master secret with its private key before it can continue the SSL handshake process.

For the client, performing the asymmetric encryption of the pre-master secret may be laborious, but it only needs to do this once per SSL session. In contrast, a web server, with many simultaneous connections, will struggle to perform the asymmetric decryption operations needed to negotiate new SSL handshakes with hundreds of new clients. These asymmetric decryptions create a processing bottleneck that degrades the server's performance—if the server is occupied performing decryption operations, it is unavailable to serve pages — causing increased client response times and limiting the server's ability to handle incoming connections.

Web server applications run as services on industry-standard hardware and operating systems, relying on the computer's processing power to perform SSL operations as required. As such, web servers are subject to the same application overhead and system resource allocation issues that affect all applications—they must share the computer's operating environment, including memory, hard disk space, and CPU processing cycles, with all other applications currently running on the computer. Couple this resource sharing with the fact that standard processors and software are designed for general computing tasks rather than specialized cryptographic processing, and it becomes apparent why dedicated cryptographic processing devices, called SSL accelerators, have an important role to fill. As web server traffic loads increase, the asymmetric calculations inherent in SSL processing quickly outstrip the capabilities of the server's CPU.



SSL Accelerators and Asymmetric Cryptographic Processing Offload

To allow web servers to effectively cope with a large volume of SSL connections, dedicated hardware-based SSL accelerators have been designed to offload asymmetric computation tasks and relieve the burden from overtaxed web servers—freeing them to serve more pages and accept more client connections. SSL accelerators may not perform symmetrical encryption operations, as these operations can be processed locally on the web server without creating undue strain on the CPU.

Instead, an SSL accelerator provides dedicated hardware acceleration for the computations required to process asymmetric cryptographic operations, in particular, the decryption of the pre-master secret exchanged during the SSL handshake.

An SSL accelerator offloads the asymmetric operations from the server, accelerating the decryption process during the exchange of the pre-master secret. In contrast to a web server's general capabilities, SSL accelerators offer a task-specific hardware solution designed to address computationally intensive asymmetrical cryptographic calculations.

Through the use of specialized processors, SSL accelerators are designed to compute large volumes of asymmetric cryptographic operations with ease. Because the ability to perform the asymmetric cryptographic computations essential to the SSL handshake is a choke point for web servers, the addition of an SSL accelerator delivers a dramatic increase in performance as the web server is relived from the strain of asymmetrical decryption.

It is important to note that a web server's ability to set up SSL sessions is reined in by a number of factors—the web server's processor speed, memory capacity, operating system, and web server software all play important roles in determining overall SSL performance. Page size, cryptographic key size, and network conditions also impact SSL performance, so it is important to maintain a holistic view of the web server's hardware, software, and content in regards to performance expectations. In many cases, an SSL accelerator's processing capabilities can outstrip those of its host web server, meaning that the SSL accelerator is not being used to its full capacity due to limitations in its host configuration. However, even in these situations, SSL accelerators provide marked gains in performance by offloading the asymmetric cryptographic processing that prevent web servers from serving pages.

Unfortunately, often lost in the dash to maximize SSL performance is the very security that SSL is supposed to offer; many SSL accelerators designed to accelerate asymmetric cryptographic operations neglect to protect the SSL private keys that are critical to a web server's identity. With the grave consequences from the loss or theft of the private keys used to identify a web server, is it possible to have strong key management without sacrificing superior SSL performance?

Unifying Hardware Key Management and SSL Acceleration

The availability of high-performance cryptographic processors on the market today make it relatively easy to create an SSL accelerator that offers performance at the expense of security. In contrast, the design, manufacturing, and operational modeling required to develop an HSM that protects key materials requires a deeper understanding of underlying issues surrounding cryptography and operational security. A truly secure SSL accelerator must rate high on both the performance and security axis to guarantee that SSL private keys remain above compromise, and to ensure that clients can continue to place trust in the identity of SSL web servers and the privacy of transactions performed during SSL sessions.

SafeNet Enterprise HSM for SSL — SSL Acceleration with HSM Security

SafeNet has drawn upon its experience designing hardware security modules (HSMs) and cryptographic processors to produce SSL accelerators that provide high-performance SSL acceleration and FIPS-validated hardware key management. With the Enterprise HSM for SSL, SafeNet has added hardware key security, performance, and increased flexibility to the SSL accelerator market. Most SSL accelerators are bound to their host web server through their dependence on the PCI-card form factor or cumbersome external bus (for example, SCSI) for connectivity to clients, severely limiting the scope and flexibility of deployment due to their direct, one-to-one attachment to their host. In contrast, the Enterprise HSM for SSL is a stand-alone network appliance that connects to web servers across a TCP/IP network with the added security of Network Trust Links that secure communication between the Enterprise HSM and web servers.

Freed from the restrictions imposed by the local bus, Enterprise HSM can be shared by authenticated web server clients that have access to a network, allowing the Enterprise HSM to be shared between multiple web servers quickly and easily—drastically reducing integration, deployment, and management overhead. The Enterprise HSM's comprehensive, multi-layer security, built-in HSM Best Practices, and integrated FIPS 140-2 Level 3-validation ensure that powerful SSL acceleration can be delivered with the same SSL private key security expected from an HSM.

To address the demands of today's high-volume, high-security SSL customers, the SafeNet Enterprise HSM for SSL offers a number of features that allow it to perform SSL acceleration while protecting sensitive SSL private keys:

High-Performance SSL Acceleration

Enterprise HSM for SSL delivers up to 6000 transactions per second (1024-bit RSA decryptions) to meet the requirements of the most demanding SSL applications.

Integrated FIPS-validated Hardware Key Management

The Enterprise HSM for SSL features a dedicated HSM, the K6 Cryptographic Engine, to perform cryptographic operations and provide secure storage for SSL private keys. The K3 Cryptographic Engine provides the Enterprise HSM's HSM functionality, offering FIPS 140-2-validated hardware-secured key management, SSL acceleration (over 6000 transactions per second), administrative access control, and policy management.

Network Shareable

The Enterprise HSM for SSL is network shareable, allowing multiple web servers to use the Enterprise HSM's SSL acceleration features concurrently, making it ideal for service providers or enterprise customers who host multiple web servers. Enterprise HSM for SSL clients are web servers that connect to the Enterprise HSM to use its HSM capabilities. Each web server communicates with the Enterprise HSM for SSL through Network Trust Links (NTLs) authenticated with digital certificates and unique client passwords.

Scalability

Multiple Enterprise HSM for SSL machines can be pooled together to scale capacity as needed.

Manageable

A compact 1U rack-mount chassis and remote administration interface make the Enterprise HSM for SSL ideal for data center environments, where space and manageability are primary concerns.

Network Trust Links — NTL

Network Trust Links (NTLs) are secure, authenticated network connections between the Enterprise HSM for SSL and its clients. NTLs use two-way digital certificate authentication and SSL data encryption to protect sensitive data as it is transmitted between the Enterprise HSM and Clients.

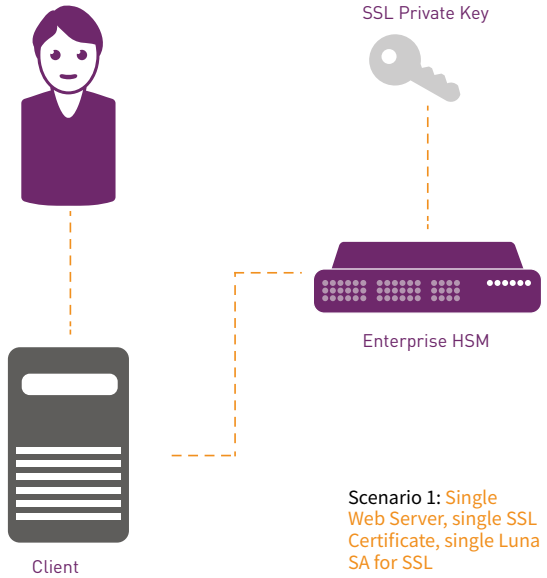
- NTLs consist of three parts:
- The Network Trust Link Server (NTLS) on the Enterprise HSM
- Network Trust Link Agents (NTLA) installed on clients
- The Network Trust Link itself, a secure connection that is created between the NTLS and an authenticated NTLA

The Enterprise HSM can support multiple NTL connections to permit many clients to share the Enterprise HSM for SSL key management and SSL acceleration capabilities.

Enterprise HSM's network connectivity, combined with support for multiple clients and multiple HSM partition configurations, allow it to be deployed in a number of different ways to suit the unique needs of different SSL web servers and server form environments.

Scenario 1: Dedicated SSL Acceleration

This is the simplest scenario, where one Enterprise HSM performs dedicated SSL key management and acceleration duties for a single web server. Because multiple Enterprise HSM's can be pooled together, if performance requirements increase, additional Enterprise HSM for SSL units can be added to the network to increase capacity.

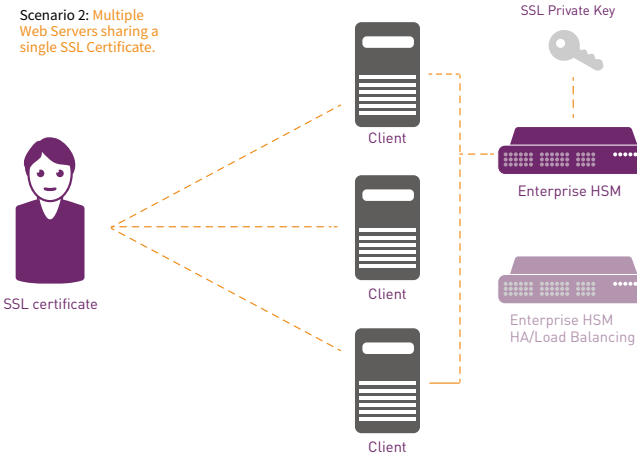


Scenario 1: Single Web Server, single SSL Certificate, single Luna SA for SSL

Scenario 2: Shared SSL Certificate

In this scenario, multiple web servers share one certificate with the related private key stored in the Enterprise HSM for SSL. Each Web server shares this certificate, and shares the Enterprise HSM for SSL's acceleration capacity. Note that each web server client can use up to 100 percent of the acceleration capacity on shared load basis—if only one client of four is active, it will receive all of the Enterprise HSM for SSL's processor capacity; if two clients are active, each will receive 50 percent.

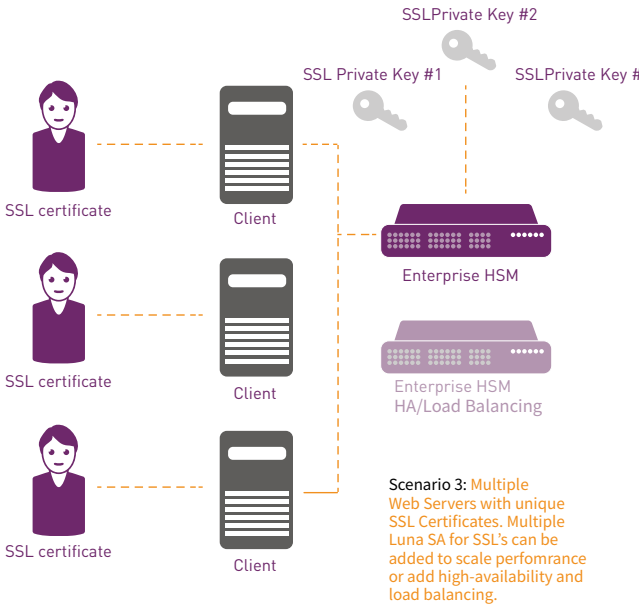
This scenario is useful in high-availability or cluster scenarios, where multiple web servers all operate under the same logical identity. Alternatively, in a hosting environment or commerce application aggregation, multiple virtual hosts residing on several web servers can securely share a single certificate



Scenario 3: Multiple Certificates with Multiple Clients

In situations where multiple web servers each maintain their own independent SSL certificates, the Enterprise HSM for SSL can provide independent key management for each certificate's private key, allowing each client to establish separate key management policies and client access controls to the certificate.

This scenario is useful in solutions where multiple distinct web servers each require their own management capabilities, such as departmental web servers maintained by common IT staff, or in the service provider space where collocated web servers all want independent SSL certificates but do not require a dedicated SSL accelerator.



Conclusion

By offering secure hardware key management and powerful cryptographic processing, the Enterprise HSM for SSL provides secure, high-performance SSL acceleration that permits the deployment of trustworthy SSL web servers. Its hardware key management mitigates the risks associated with weaker, software-based key management, protecting SSL keys with FIPS 140-2-validated hardware that is impervious to both physical and electronic attacks that would cause lesser solutions to surrender sensitive keys to exploitation. In addition, Enterprise HSM's dedicated cryptographic engine allows it to deliver high-speed SSL acceleration to free web servers from processor-intensive SSL handshakes that cause secure websites to grind to a halt.

Couple these features with the Enterprise HSM's unmatched network deployment flexibility that allow it to easily adapt to a range of different SSL deployment scenarios, ranging from a single web server with a sole SSL private key, to 20 web servers, each with its own dedicated SSL private key, and the Enterprise HSM proves that it is possible to provide high-performance SSL acceleration without placing security in jeopardy.

Contact Us: For all office locations and contact information, please visit www.safenet-inc.com
Follow Us: data-protection.safenet-inc.com

GEMALTO.COM

gemalto
security to be free