ISSN:0975-9646

An Enhanced Approach to Use SSL for End To End Security

Swati Gupta, Saru Dhir

Amity School of Engineering and Technology
Amity University, Noida

Abstract— **As today security is highly dependent on SSL especially in the field of e-commerce, it has become an important area of research to improve speed, availability of services, security etc. Here in this work the focus is on making it work more efficiently even when connection to the CA (Certificate Authority) is not available. This paper is focused on enhancing the speed and availability of the existing model which is used in the SSL. In SSL basically three components are involved:-Client, Server, CA. CA plays an important role in the SSL (Secure Socket Layer) authentication because it provides certificates to clients and servers on which both client and server trust. When a request is send to the server, server verifies the client by verifying its certificates with the CA and CA validates the certificates. So if in case when the connection to CA is not available than server cannot verify the certificates and connection will not be established which will prevent client from accessing the services even if it is an authenticated client. In this paper, the proposed model will deal with this problem, the proposed model will have a local CA node at every client and server which will be always in sync with the CA so that when a request comes from the client server can easily verify its certificates using local CA node.**

Index Terms—SSL(Secure Socket Layer), CA (Certificate Authority), security, DNS.

I. Introduction

Originally deployed in Web browsers, SSL has become standard for secure Internet communications. The main function of SSL is to provide end-to-end security against a man-in-the-middle attacker[3]. Even if the network is completely compromised—DNS is poisoned, access points and routers are controlled by the adversary, etc.

SSL is intended to guarantee confidentiality, authenticity, and integrity for communications between the client and the server. Authenticating the server is a critical part of SSL connection establishment. This authentication takes place during SSL handshake, where the server presents its public-key certificate[2]. In order for the SSL connection to be secure, the client must carefully verify that the certificate has been issued by a valid certificate authority, has not expired, the name listed in the certificate match the name of the domain that the client is connecting to, and perform several other checks. As shown in Fig.1. a client is connecting to the server over SSL so client sends the hello message to the server and server response with the acknowledgement message by sending it certificate. Client then verifies the certificate and if certificate are valid certificate then client exchange a key with the server and a secure connection is build and now all the information is shared between client and server is encrypted.

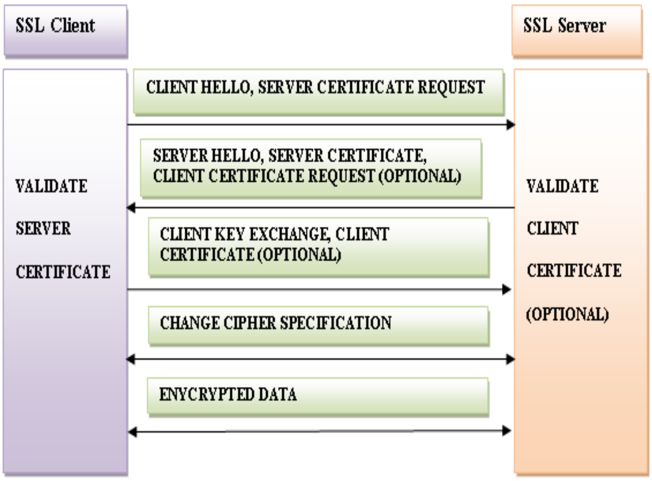
SSL Client
SSL Server
VALIDATE SERVER CERTIFICATE
VALIDATE CLIENT CERTIFICATE (OPTIONAL)
CLIENT HELLO, SERVER CERTIFICATE REQUEST
SERVER HELLO, SERVER CERTIFICATE, CLIENT CERTIFICATE REQUEST (OPTIONAL)
CLIENT KEY EXCHANGE, CLIENT CERTIFICATE (OPTIONAL)
CHANGE CIPHER SPECIFICATION
ENYCRYPTED DATA

Fig. 1. Simplified overview of SSL handshake

II. Overview Of SSL

SSL provides endpoint authentication and encryption over the internet. It is widely used for secure communication, especially in the area of e-commerce. SSL begins with a handshake between the client and server to perform authentication and to establish a session key. Server authentication is performed with the use of X.509 certificates. Authentication relies on components in the browser, CA, and DNS to operate seamlessly. A client authentication typically depends on passwords. SSL is the secure communications protocol of choice for a large part of the Internet community[7]. There are various applications of SSL in existence, since it is capable of securing any transmission over TCP. Secure HTTP, or HTTPS, is a familiar application of SSL in e-commerce or password transactions[9]. According to the Internet Draft of the SSL Protocol, the point of the protocol "is to provide privacy and reliability between two communicating applications." The protocol further explains three points combine to provide connection security[8]. These points are:

- Privacy connection through encryption
- Identity authentication – identification through certificates, and
- Reliability –dependable maintenance of a secure connection through message integrity checking.

A. *Components in SSL*

- Client
- Server
- Certificate Authority(CA)

B. *What is Certificate Authority?*

A Certificate Authority is an entity which issues digital certificates to organizations or people after validating them. Certification authorities have to keep in-depth records of what has been issued and the information used which is used to issue it, and these are audited regularly to make sure that they are following predefined procedures. All certification authority provides a Certification Practice Statement (CPS) that defines the procedures that will be used to verify applications.

III. SSL AUTHENTICATION

The SSL authentication process uses certificates that are issued by a certificate authority and the same process applies if the certificates are issued by an certificate generation utility or if self-signed certificates are used.[4]

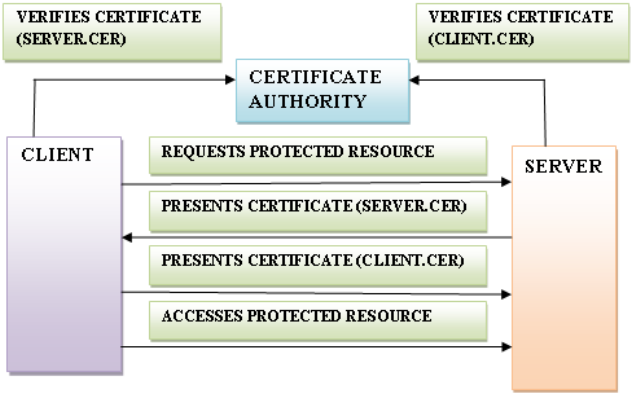
VERIFIES CERTIFICATE (SERVER.CER)
VERIFIES CERTIFICATE (CLIENT.CER)
CERTIFICATE AUTHORITY
CLIENT
SERVER
REQUESTS PROTECTED RESOURCE
PRESENTS CERTIFICATE (SERVER.CER)
PRESENTS CERTIFICATE (CLIENT.CER)
ACCESSES PROTECTED RESOURCE

Fig. 2. Authenticating the identity of an application

To establish an SSL connection:

- A SSL client application contacts an application which is acting as an SSL server.
- The SSL server replies by sending the signed certificate issued by CA to the SSL client. Certificate contains identifying information about the CA which has issued the certificate and the application that presents the certificate, digital signature of the CA, and a public key.
- The SSL client than verifies the certificates with the CA.
- After verifying the signature on the certificate, the SSL client requests the SSL server to verify its identity.
- The SSL servers use its private key to encrypt a message.
- The SSL server sends that encrypted message to SSL client.
- To decrypt that message, SSL client will use the public key embedded in the signed certificate it received, and there by verifies the identity of the owner of the certificate.

A. *Types of SSL authentication*

- *One way SSL authentication:* One-way SSL authentication allows the application operating as the SSL client to verify the identity of the application operating as the SSL server.
- *Two way SSL authentication:* In two-way SSL authentication, SSL client application verifies the identity of the SSL server application and then SSL server application verifies the identity of the SSL-client application[5].

B. One way SSL authentication:

In SSL authentication, the client is presented with a server's certificate and the client computer might try to verify the server's CA against the client's list of trusted CAs. If the issuing CA is in trusted list, then the client will verify that the certificate which it receive is authentic and it has not been tampered with. During this aspect, both client and server use many handshake messages to establish the encrypted channel prior to message exchanging. These messages are:-

- Client sends a Hello message to the server proposing SSL options.
- Server shows response with server Hello message and selecting the SSL options.
- Server sends its Certificate message, which contains the server's certificate.
- Client verifies the certificates with the CA.
- Client sends session key information (encrypted with server's public key) in Client Key Exchange message.
- Client then sends a change Cipher Specification message to activate the negotiated options for all future messages.
- Client then sends a Finished message to allow the server to check the newly activated options.
- Server then sends change Cipher Specification message to activate the negotiated options for all future messages.
- Server then sends Finished message to allow the client check the newly activated options.

C. Two way SSL authentication:

Mutual SSL authentication or certificate based mutual authentication refers to two parties authenticating each other through verifying the provided digital certificate so that both parties are assured of the each others' identity.

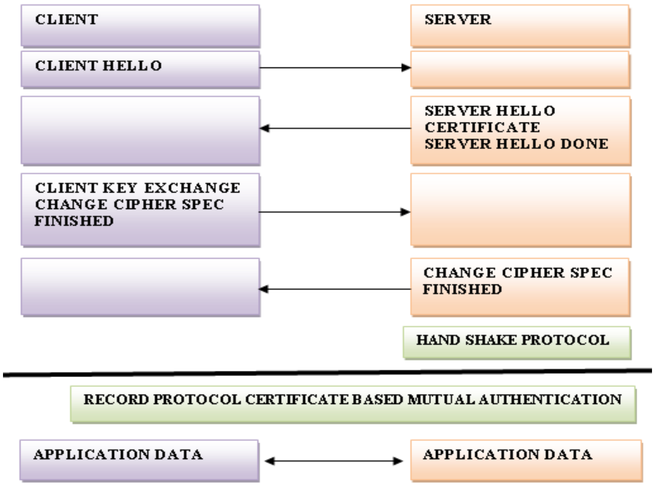

Fig. 3. SSL authentication handshake messages

In Technical terms, it refers to a client (web browser or client application) authenticating themselves to a server (website or server application) and that server also authenticating itself to the client through verifying the public key certificate/digital certificate issued by the trusted Certificate Authorities (CAs). Because authentication relies on digital certificates, the certification authorities like Verify sign or Microsoft Certificate Server are an important part of the mutual authentication process. At a high-level point of view, process of authenticating and establishing an encrypted channel using certificate-based mutual authentication involves the following steps:

- Client sends Hello message to the server proposing SSL options.
- Server replies with server Hello message and selecting the SSL options.
- Server sends its Certificate message, which contains the server's certificate.
- Server requests a client's certificate by certificate Request message, so that connection can be mutually authenticated.
- Client than verifies with the CA.
- Server completes its part of the negotiation with server Hello Done message.
- Client responds with message that contains certificate, which contains the client's certificate.
- Server verifies the certificates with the CA.
- Client sends session key information (encrypted with server's public key) in Client Key Exchange message.
- Client sends a Certificate Verification message to let the server know it owns the sent certificate.
- Client then sends change Cipher Spec message to activate the negotiated options for all future messages.
- Client then sends Finished message to allow the server check the newly activated options.
- Server then sends change Cipher Spec message to activate the negotiated options for all future messages.
- Server sends Finished message to allow the client check the newly activated options.

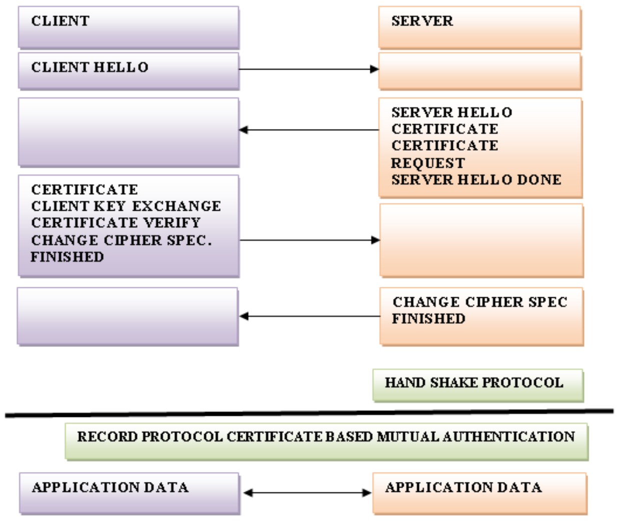

Fig. 4. Mutual SSL authentication handshake messages

IV. Basic Services Of SSL

- Message privacy is achieved through a application of public-key and symmetric-key encryption. All the traffic between an SSL client and an SSL server is encrypted by using a public-key, and encryption algorithm is negotiated at the time of session setup[1].
- Message integrity ensures that SSL session traffic does not corrupt the contents of the message while it is on its way to its final destination. SSL uses a combination of public keys and private keys, and hash functions so that message integrity is assured.
- Mutual authentication is the process whereby the client and the server convince each other of their identities. The client and server identities are encrypted in public-key certificates. A public-key certificate mostly contains the following components:
    - Subject’s distinguished name
    - Issuer’s distinguished name
    - Subject’s public key
    - Issuer’s signature
    - Validity period
    - Serial number

For server-side authentication, the client must also have access to the certificates of the authorities that the client trusts. In general, client's key ring can contain either the certificates (without the corresponding private key) of the servers that the clients can trust, or the certificates of the authorities that the clients can trust. The key ring which is at the client must be paired with the key rings that are at the servers that the client uses. Similarly, for client authentication, the server requires access to those certificates of the CAs that have been used to generate client certificates.

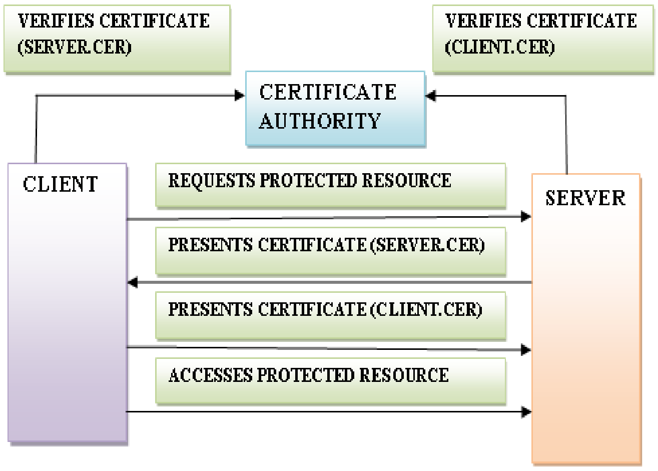
VERIFIES CERTIFICATE (SERVER.CER)
VERIFIES CERTIFICATE (CLIENT.CER)
CERTIFICATE AUTHORITY
CLIENT
SERVER
REQUESTS PROTECTED RESOURCE
PRESENTS CERTIFICATE (SERVER.CER)
PRESENTS CERTIFICATE (CLIENT.CER)
ACCESSES PROTECTED RESOURCE

Fig. 5. Existing Model

V. Working Of Existing Model Of SSL

- Client sends request message to the server using SSL
- Server responds and request for the certificates.
- Client responds with its certificates request for the server certificates
- Server verifies the certificates with the CA.

- CA verifies the certificates.
- Server responds to the client with its certificates.
- Client verifies it with the CA.
- Finally a secure and trusted connection is setup[6].

VI. Proposed Model

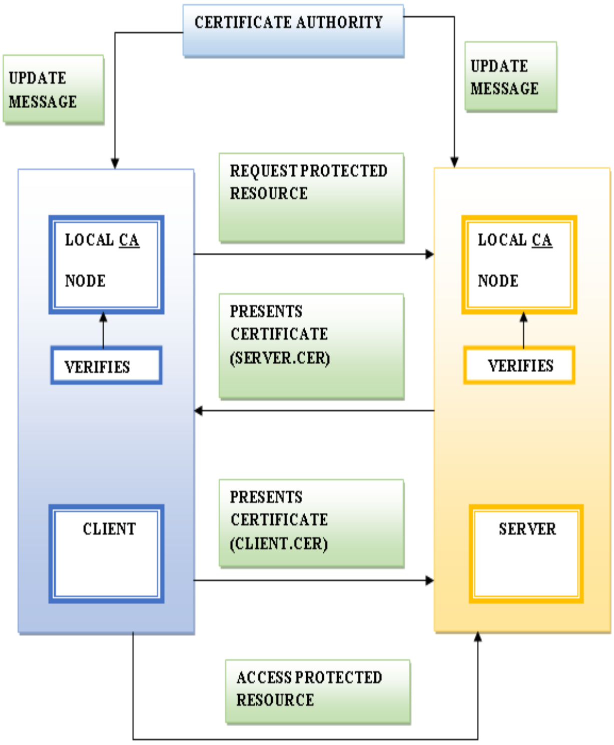
CERTIFICATE AUTHORITY
UPDATE MESSAGE
UPDATE MESSAGE
REQUEST PROTECTED RESOURCE
LOCAL CA NODE
VERIFIES
LOCAL CA NODE
VERIFIES
PRESENTS CERTIFICATE (SERVER.CER)
PRESENTS CERTIFICATE (CLIENT.CER)
CLIENT
SERVER
ACCESS PROTECTED RESOURCE

Fig. 6. Proposed Model

A. *Components of Proposed Model*

- *Client:* clients is a system which requires to connect to a server over SSL to access some services.
- *Client local CA node*:-Its replica of CA which is in sync with the CA and contains all information about the certificates. It keeps getting updated every time a new certificate is issued or an older certificate is revoked. These nodes only have to sync with the CA when CA issues a token that a new certificate has been issued or an older certificate is revoked. So no overhead is created.
- *Server:*-server is a system which satisfies the request of the client
- *Server local CA node*:- Its is replica of CA which is in sync with the CA and contains all information about the certificates . it keeps getting updated every time a new certificate is issued or an older certificate is revoked. These nodes only have to sync with the CA when CA issues a token that a new certificate has been issued or an older certificate is revoked. So no overhead is created.
- *CA:*-CA is certificate issuing and verifying authority**.**

B. *Working of local CA model*

- Local CA node is always in sync with the CA.
- All the certificates issued by the CA are available with the local CA node. Whenever CA issues any new certificate it broadcasts an update message to all local CA nodes which gets synchronized with the CA.
- Update message contains the new certificate so that local CA node need not to search for the new certificate, it just simply add that to their trusted list.
- Whenever CA revokes any older certificate it broadcast an update message to all local CA nodes which gets synchronized and removes that certificate from trusted list.

C. *Working of Proposed Model*

- Client sends request message to the server using SSL.
- Server replies with a request of client's certificates.
- Client sends its certificates and requests server's certificates.
- Server verifies the Client's certificates with the local CA node.
- Server replies with its certificates and completes its part.
- Client verifies server's certificates with the local CA node.
- Client completes its part by sending acknowledgement message.
- Secure connection is than established by any decided encryption technique.

VII. Features And Benefits Of Proposed Model

- As local CA node has to only sink when CA broadcast a message, so no message overhead is created.
- Also now client and server need to connect only to local CA node so verifying process is fast.
- Now client and server need not to be connected to CA all the time so their workload and overhead is reduced.
- No external network is accessed in verifying process so the process becomes more secure.
- Also if CA is not available or connection to CA fails system can work using its local CA node.

VIII. Conclusion

Due to huge dependency on SSL it has become an important area of research and lots of scientists are putting their lots of efforts to improve the speed, availability and performance of an SSL system. So in this work our main area of concern is to make this system work more efficiently. Our main focus is to make the system work even when the connection to the CA is not available. This will enhance the speed and availability of the current model.

Proposed model will work on the fact that at every server and client there will be dedicated CA node. The local CA node will be responsible to remain in synchronization with the certificate authority. Whenever a request comes, certificates are verified from the local CA node. So this will improve the speed because the local CA node is in the local network of the client and server. It will also improve the security as to validates certificate no external network is accessed.

REFERENCES

[1] Yong Song, Konstantin Beznosov, Victor C. M. Leung " Multiple-channel security architecture and its implementation over SSL" in EURASIP Journal on Wireless Communications and Networking Volume 2006 Issue 2, April 2006.

[2] Giacomazzi, Poli, A. " Cost-Performance Optimization of SSL-Based Secure Distributed Infrastructures"in Latin America Transactions, IEEE (Revista IEEE America Latina) Volume: 9 , Issue: 4 , Page(s): 550 – 556 in 2011.

[3] Kambourakis Rouskas, A. ; Kormentzas, G. ; Gritzalis, S. "Advanced SSL/TLS-based authentication for secure WLAN-3G interworking " in IEEE preceding Volume: 151 , Issue: 5 ; pages): 501 – 506 in 2004.

[4] C. J. Lamprecht, Aad and P. A. van Moorsel "Adaptive SSL: Design, Implementation and Overhead Analysis " in IEEE Computer Society, July 2007.

[5] Jon Howell, David Kotz "End-to-end authorization " in OSDI': Proceedings of the 4th conference on Symposium on Operating System Design & Implementation - Volume 4, in USENIX Association October 2000**.**

[6] Norman Lim, Shikharesh Majumdar, Vineet Srivastava "Engineering SSL-based systems for enhancing system performance" in Proceedings of the 2nd ACM/SPEC International Conference on Performance engineering, March 2011.

[7] Kapil Singh, Helen J. Wang, Alexander Moshchuk, Collin Jackson, Wenke Lee "Practical end-to-end web content integrity" in Proceedings of the 21st international conference on World Wide Web in ACM, April 2012.

[8] Martin Georgiev, Subodh Iyengar, Suman Jana, Rishita Anubhai, Dan Boneh, Vitaly Shmatikov "The most dangerous code in the world: validating SSL certificates in non-browser software " Proceedings of the 2012 ACM conference on Computer and communications security ,October 2012.

[9] Light, J. Ikejiani, O.K. "An efficient wireless communication protocol for secured transmission of content-sensitive multimedia data " in World of Wireless, Mobile and Multimedia Networks & Workshops IEEE Page(s): 1 – 6, April 2009.