Synchronize Confidential Authentication (SCA)

Mazen AlObaidi [A*], Huirong Fu[A], and Ye Zhu[B]
Department of Engineering and Computer Science
[A]Oakland University
Rochester, Michigan, USA
[B]Cleveland State University
Cleveland, OH
Malobaid@Oakland.edu, fu@oakland.edu, zhuye06@gmail.com

Abstract

Digital identity management is one that is becoming an integral part of our lives, as consumers and businesses rely more and more on online transactions for daily tasks, such as banking, shopping, and bill payment. These transactions crucially depend on networked computer systems to communicate sensitive identity data across personal, company, and enterprise boundaries. Like regular offline transactions, identity theft is always a threat. And it is known that in the digital world, there is nothing that is 100% secure. In this paper, we propose Synchronize Confidential Authentication model that resembles the real world notary service only to become a part of online transactions authentication through a trusted third party referred to as Certificate Authority whose responsibilities are clearly defined. Our proposed model binds confidential pair of keys to remote user/server, no key exchange algorithm needed and presents a new protocol for mutual identity authentication using client authentication flow with variations in its implementation from existing models in the literature that can be summarized in two points: using generation of related pair of confidential keys for user and server instead of using the public key in the client authentication flow and re-generation of new confidential keys based on a time interval as requested by the requestor from Certificate Authority.

Introduction

The fast increasing number of web services is transforming the web from a data oriented repository to a service oriented repository. A continuous concern when using the web is the issue of security and protecting the information being transmitted on the web. Many solutions have been designed to prevent some malpractices from taking place. The main building block of security is Authentication [1]. In theory, authentication is relatively simple: A user provides some sort of credentials—a password, smart card, fingerprint, digital certificate—which identifies that user as the person who is trusted to access the system. There are, however, varieties of methods and protocols that can be used to accomplish this. Nevertheless of the method, the basic authentication process remains the same [2]. In this paper we are proposing a novel solution that aims to meet authentication, and automated identification. In addition, it is an extended digital certificate model. Digital certificate is a binary file that is used for authentication and securing of communications. Certificate Authority associates a public key to an entity (a user, a computer or service) that has the corresponding private key. In addition, the digital certificate has potential security vulnerability in using asymmetric keys is the possibility of a "man-in-the-middle" attack [3]. Our proposed solution will prevent man-in-the-middle attack. Since our proposed model is online then the renew certificate is automated and this will reduce shortsighted management practices [4]. This paper is structured as follows: Section 1 presents related work. In section 2, we outline the framework of Synchronize Confidential Authentication. In section 3, we describe the future works. In section 4, we present the conclusion.

1. Related Works

Effective solutions have been developed to solve the identity theft problem in web services. One well-known identity management solution that deals with this issue is the single sign-on (SSO) technique, which requires the user to authenticate only once to a website, and then automatically authenticates the user to other websites from then on, within a session. The approach based on cryptographic-enabled assertions is embodied by protocol Security Assertion Markup Language (SAML) [5]. This solution has performance issue since the internet is stateless, therefore the single sign on software must check every request. In [6], the Security-token is technique to strength the password authentication by using hardware device that has an algorithm to generate random number that the security server uses it to authenticate the user. The vulnerability of this solution is when the key fob is stolen.

2. SCA Model

A main requirement to implement the model described above is to complete a registration process. This registration comprises both the user and the server as in Figure 1. The registration for each is asynchronous. A user can register at any time before initiating a service request. A server can register at any time before or at the time of the user registration. The trusted third party is responsible to ensure that once the user registers that the server is registered or should be contacted for registration. To state the responsibilities of the trusted third party, we make the following assumptions:

1) Trusted third party is a Trusted organization
   issue confidential certificates aka CA
2) Generating pair of confidential key
   using asymmetric key cryptography
   a. User confidential key
   b. Server confidential key
3) Generating Confidential Certificate (CC)
4) Send the Confidential Certificate over
   a secure channel to user and server
5) Support real time Identification

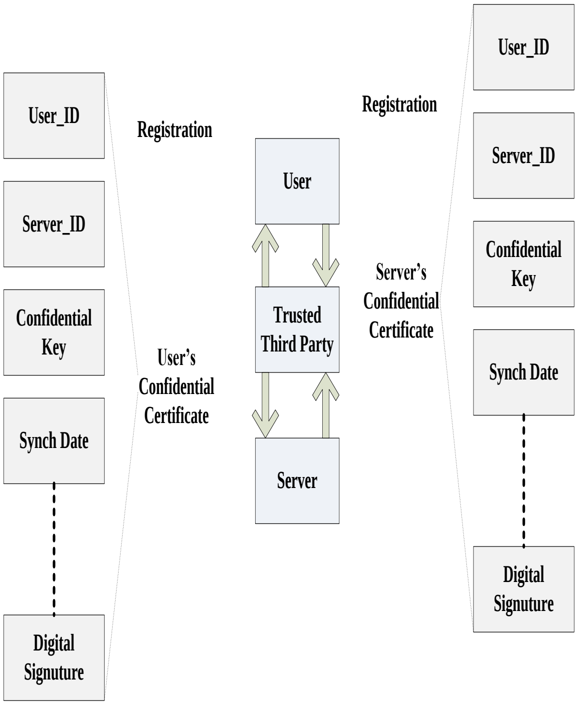

Figure 1: Registration Process with Trusted Third Party (CA)

A secure channel is a requirement for a successful implementation of our model to prevent any malicious attacks such as wrapping attack or malware-infection attack to intercept the transmitted message and temper with its content or its sender identification.

2.1 SCA protocol

SCA protocol is an extension of SSL Protocol version 3.0 that was released by Netscape in 1999. Our proposed protocol will extend the SSL to accommodate/service our proposed model. While SSL does not require mutual authentication, the SCA protocol requires that they both have a CC. The SSL protocol contains two main layers: Record and Handshake layer.

The first layer is Record layer where all SSL protocol messages move in records of up to 32,767 bytes. Each message has a header of either 2 or 3 bytes [7].
The second layer is Handshake layer that contains three sub-layers as follows:

1) Handshake: Allowing the server and client to authenticate each other and negotiate an encryption and cryptographic keys to be used to protect data sent in an SSL record
2) Change Cipher Spec: The Change CipherSpec layer signals the beginning of secure communications between the client and server and the changing in communications protocol
3) Alert: This protocol sends errors, problems or warnings about the connection between the two parties

In SCA protocol we propose in Figure 2, a new sub-layer into the Handshake layer for synchronization, we call it the Synch layer. This synchronization, we referred to earlier, handles the new key generation during the key synch interval.

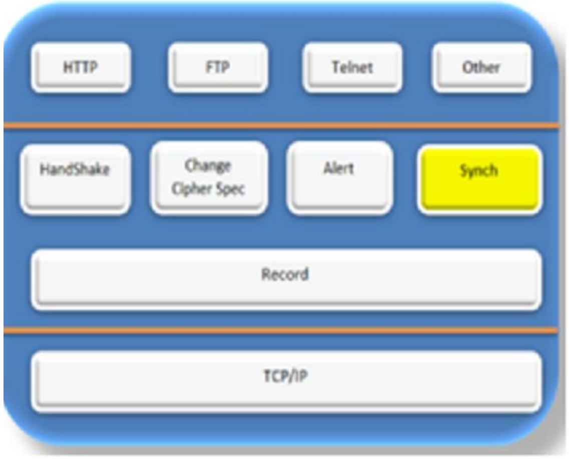

Figure 2: SCA Protocol based on SSL

The Synch layer provides the following services:

1. Reading the CC and setting the synchronization value of the SSL
2. Terminating the session when the synchronization occurs
3. Sending request to CA for generating new pair of confidential keys
4. Receiving the new confidential key from CA
5. Updating CC with new confidential keys
6. Sending message to Alert layer when the synchronization occurs
7. Sending message to invoke Changing cipher Spec

3. SCA Paradigm

Our solution has a broad scope of use in the world of enterprise's application. It can be used for any online transaction with enhanced security measures and identity authentication using Confidential Certificate. Certificates are issued after rigorous authentication will be more trustworthy than certificates requiring little or no authentication. We propose a model that authenticates both sides of the transaction as well as any or every request from either side by satisfying all the security requirements listed above. Our model complies with the key distribution and key management techniques that a CA is responsible for. However, we present a variation in modeling the digital certificate issued by a CA. A precondition to initiate a session as described above between user and server is the registration of one or both of them with the third trusted party as described in details above. Once registered, both user and server can participate in an online transaction as described in the diagram in Figure 1 step by step. The trusted third party authenticates the identity of both user and server and supplies them with a confidential certificate that contains a confidential key to be used for authentication, decryption, or digital signature. As the schema in Figure 3 is illustrates the view of the architectural model of our proposal approach. The algorithm will be automatically executed once the synchronization interval expires so new keys are generated to resume the request. Otherwise, the request will be rejected by the server. Each server will have a different key for each user for each channel.

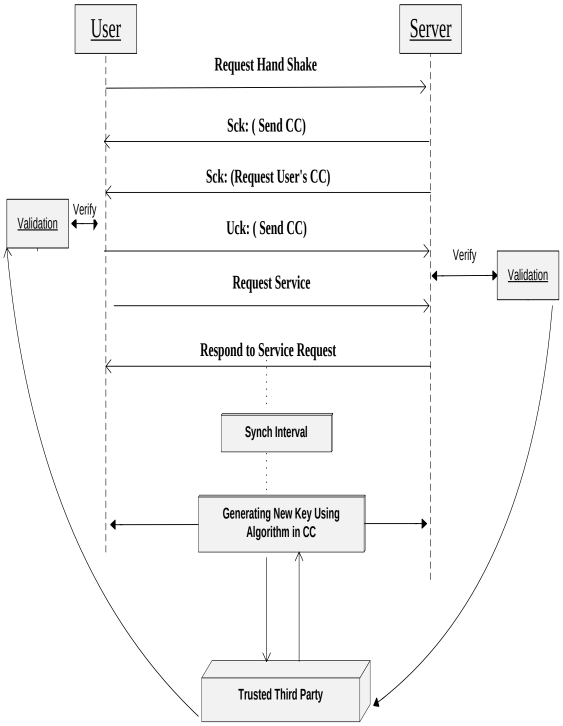

Figure 3: Synchronize Confidential Authentication as a Service Diagram

Since the keys are symmetric keys and not shared between the user and server, then they can service as an authentication to each of the parties. This model provides the following:

1- Mutual Authentication of both User and Server

2- Privacy by using confidential key to encrypt and decrypt the data

3- Integrity by using confidential key to genertae digital signeture

4-Generation of new confidential key with/without the need for the trusted third party after the Activation/registration process.

5- An attempt to formulate a new algorithm that generates a pair of Confidential keys that can encrypt and decrypt each other’s messages

Here is a typical sequence of steps when applying the SCA:

1. The user initiates handshake request with server
2. Server responds by send its CC & request user's CC
3. The Server removes his confidential key from the CC, encrypts it with his confidential key, and sends it to User
4. User decrypts the server message with his own confidential key and gets server's CC
5. User validates the server's CC with CA
6. User removes his confidential key from the CC, encrypts the CC, and sends it to server
7. Server decrypts user message with his own confidential key and gets the user's CC
8. Server validates the user's CC with CA
9. For the duration of the SSL session, the SSL server and SSL client can now exchange messages that are encrypted.
10. When the synch is activated based on the CC then the Synch protocol will send request for CA to get new CC when synch intervals expires so new keys are issued
11. Start from step 1

4. Future Work

As a future work in continuance of this solution, we would implement a proof of concept of our proposed model and compare it with the currently used DC framework to determine the benefits of having SCA as an improvement. Also, implementing the local SCA type which we believe that might eliminate the trusted third party and improve the performance and security. Furthermore, we would like to experiment a new feature we call "Imitation request" that will prevent play back attack, and active intruder

5. Conclusion

Nowadays, web activities are popular and consist of more than just looking for information but it also provides the opportunity of purchasing any kind of legal goods. This raises the concern of information confidentiality as well as the digital identity of the parties involved in any transactions. In this project, we propose a model for information security that addresses several concerns by using secure communications protocol. Authentication is the main purpose of issuing a confidential certificate. SCA is a hybrid framework that comprises mutual authentication and Public Key encryption. Furthermore, SCA framework allows a dynamic request to obtain a new confidential certificate.

REFERENCES

[1] Kizza, Joseph Migga. *Computer network security and cyber ethics*. McFarland, 2011.

[2] Franks, John, et al. "RFC 2617: HTTP Authentication: Basic and Digest Access Authentication." *Internet RFCs* (1999).

[3] http://www.nrc.gov/site-help/e-submittals/faqs/intro-auth.html
[4] http://www.baselinemag.com/security/the-hidden-threats-of-security-certificates
[5] https://www.oasisopen.org/committees/tc_home.php?wg_abbrev=security
[6] http://www.authenticationworld.com/Token-Authentication/
[7] http://www.homeport.org/~adam/ssl.html
[8] http://msdn.microsoft.com/en-us/library/windows/desktop/aa380513%28v=vs.85%29.aspx