**Security in E-Commerce by Implementing Digital Certificate**

In this modern era, internet is widely used all over the world. Majority of developing and modern countries have access to internet. It opens the possibility of interaction between people or organizations from different countries. One of the most interesting developments of internet is the opening of Online Shopping System, where User can buy things that are offered in a website. Everyone who has access to internet, and visit the particular online shop is a customer, and it’s not limited to only one country. This is one magnificent implementation of internet.

While it may sound handy, if you plan to open your own online shop, you should ask yourselves; “How do we proof our identity to Users?”, “How do the Users verify us?”, “How do we secure their information in the middle of transaction?”

One way of solving these issues is to get a Digital Certificate to our website. Digital Certificate is a unique electronic signifier issued by a Certificate Authority that functions like a passport to verify a user’s identity. Digital Certificate is a proof that our system has been verified by a trusted third party. From a digital certificate, a customer will be able to verify our identity as well. Therefore, they can decide whether to give our system their trust. A digital certificate is issued by an authority, referred to as a certification authority (CA). Because a digital certificate is issued by a certification authority, that authority guarantees the validity of the information in the certificate. Also, a digital certificate is valid for only a specific period of time [1]. Certificates are an essential component of any infrastructure to support digital signature [2].

**How does it work?**

To prevent our system from a potential threat, we have to support our system with several techniques. One of the available techniques is to implement Public Key Infrastructure (PKI). PKI is the basis of security infrastructure whose services are implemented and provided using public key. In a PKI, there is one entity essential for it to work, which is known as Certifying Authority (CA). CA issues a digital certificate signed by their private key [3]. When a digital certificate is published by a CA to the requesting party, the digital certificate delivered and associated with a key, which is commonly known as a public key, and of course the digital signature of the CA which is signed with the CA’s private key. The signature binds the public key with the requesting party, and the public key can be used to verify that a public key belongs to an individual [4]. All data transmitted from the system that is bound with the certificate will be encrypted with the public key, therefore, securing all transmission under the certificate. The public key can be distributed widely, as it will be used to encrypt data, but cannot be used to decrypt the data. This method is known as asymmetric-key encryption. This technique uses a key-pair to encrypt and decrypt a message, meaning it uses different keys to encrypt and decrypt the message. The signature of the CA itself is signed with the CA’s own private key, which should never be distributed.

We have discussed how to securely encrypt the information before sending it to the server. There is one other possible doubt in this case. How do we make sure that the information sent and received is exactly the same? Since an eavesdropper can’t see the content of the information sent to the server, they might still be able to alter the information, leading to incorrect information. In order to tackle this challenge, we have to verify that the information received by the server is originated from the web page, and not altered. To verify this, there is an algorithm known as MAC, which will be discussed below.

**Elements of Digital Certificate**

A digital certificate must consist of:

1. **Name of the certificate holder**

The name of person or organization who owns the digital certificate. This element is important as it shows our identity. Customer can also verify our identity by having this information in the digital certificate. [5]

1. **Serial Number of Digital Certificate**

A set of numbers that uniquely identify a digital certificate. This is important to distinguish between certificates that revoked or not.

1. **Name of Certification Authority who issued the certificate**

The name of trusted third-party who signed the certificate. A certificate must contain this information to who Customer that the certificated has been verified and trusted by this Certification Authority [5].

1. **Validity Period of the Certificate**

This information will show the period of validity of a certificate. A certificate which period has is expired should no longer be trusted.

1. **Public Key of Certificate Holder**

The information of a Public Key that is associated with the owner of a certificate. This Public Key will be used to encrypt important information while sending information through network [5].

1. **Certificate’s Signature Algorithm Identifier**

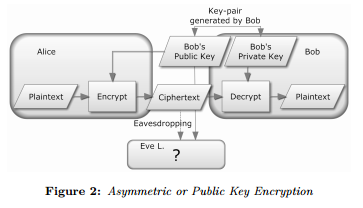
Specify the name of algorithm used by the CA’s private key to sign the digital certificate

1. **The Digital Certificate**

This is the most important element of a digital certificate, as it proof that the CA has verified the Certificate Holder’s identity [5].

**Verifying Integrity with Digital Signature**

Using this technique is a way to ensure that a message sent and received is exactly the same. The application of this method is using asymmetric-key encryption. In asymmetric key encryption (Public Key Encryption), instead of using one secret key for both encryption and decryption, this technique uses a key-pair consisting of a public key for encryption, and private key for decryption [6]. In this case, the sender creates a hash by applying the hash algorithm to the original message, hence generating an encrypted summary of the message. The hash of the message acts as a "digital fingerprint" of the message.



**Figure 1:** Encryption process [6]

When the original message changed even a little bit, the resulting hash would also change, since the hash is originated from the actual message. The sender then encrypts the hash with their private key. it is fair to say that the encrypted hash message acting as the digital signature of the message. The sender can add the digital signature at the end of its original message, and send them out. The receiver can then decrypt, only the digital signature, using the public key of the sender. From here, the receiver will get the hash of the message, by decrypting the digital signature. Next step is, applying the exact same hash algorithm to the message they received. The receiver can then compare the result of his hash algorithm with the decrypted digital signature. If the hash values are identical, the receiver can be certain that the message has not changed since the sender signed the message. If the values are different, then it is possible that the content of the message has been altered or changed by an unauthorized party, or the person who sent the message is not the person who they claimed to be. This is a way to verify that a message sent should exactly match a message received [7].

**Verifying Integrity with MAC**

Message Authentication Codes (MACs) apply symmetric cryptographic algorithms, which provide data integrity and authentication of data origin. Data integrity provides the recognition of any modification or manipulation of the message during transmission. Authentication of data origin means the confirmation that the message originates by the sender, who shares the used secret key with the receiver [8].

**Verifying the Validity of Digital Certificate**

When one needs to check the validity of a Digital Certificate, they can check the Certificate Revocation List (CRL). CRL lists the serial numbers for certificates that have been revoked, and therefore, entities presenting those (revoked) certificates should no longer be trusted [9]. The term ‘revoked’ here means that the particular digital certificate might have been compromised, or the private key of the digital certificate has been stolen. Any future signatures made from a revoked digital certificate won’t be valid. Once a digital certificate compromised, the CA who issued the digital certificate should revoke the certificate [10]. A CRL is an important entity, as it helps anyone to determine whether a certificate is revoked or not. When a serial number of a certificate listed in the CRL, any system who presents a particular certificate with those serial numbers should not be trusted.

Another option to check the validity of a Digital Certificate is to use The Online Certificate Status Protocol (OCSP). It enables application to determine the status of a known certificate, whether it is revoked or still valid. An OCSP client sends a request to OCSP responder, and holds the acceptance of the certificate until the responder provides a respond [11]. There are three kinds of answers possible as a respond:

- **good**: The certificate has been revoked by the time the request has been received. [12]

- **revoked**: The certificate has been temporarily or definitively revoked. [12]

- **unknown**: The server does not have enough information about the certificate. [12]

**Example Scenario**

A user access an online shopping website page that has a certificate attached to it. When the user visits a page, the server will display information and also send their public key. When User enters his/her information, the website will encrypt those data with the given public key, and send it to the network. Therefore, the information is not visible to other user. Let’s say there is an eavesdropper intercepts the information. The information he stole can’t be deciphered, as the private key needed to decrypt the information is securely stored in the server. Even though the eavesdropper has access to the server’s public key, they still won’t be able to decrypt it. The main point here is; the information entered by the User will be encrypted before being sent to the server, and only the server who holds the pair private key can decrypt the information.

Now the web page can also apply a digital signature to the message before sending it to the server. It creates a hash by applying hash algorithm to the original message, encrypt the hash with its private key, and finally send the encrypted message along with the digital signature. When the server received the message, server will decrypt the digital signature with the web page’s public key; therefore get the hash from the web page. Server will apply the same hash algorithm to the message sent by the web page and compare the result. If the value is identical, that means the message is not compromised. From here, the server can proceed with the transaction procedure, and so on.

**Implementation and Testing**

When implementing the sample system, for imitating the digital signature security feature in e-commerce, the major consideration was given to the interaction between the Site Owner and Certification Authority, in order to obtain a digital signature, or a certificate.

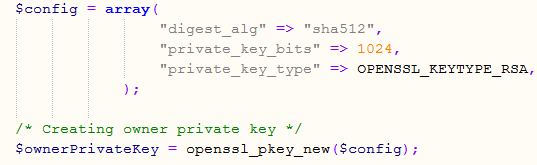
Attempts to resolve problem of the scalability and distribution of keys between participants in communication have resulted in these asymmetric-key systems. In this mechanism both site owner and CA will have a key pair – public key and private key, where data encrypted by private key is decrypted by the public key and vice versa.

In the process of digital certificate signing feature, currently implemented in the system under consideration, a private and a public key pair will be generated by the system for both the parties (site owner and CA). For generating the private key and extracting the public key from it, OpenSSL library functions have been utilized.

**Private and Public Key Generation.**

In order to generate the private key for each user, following configuration was applied.

$config = array( "digest\_alg" => "sha512", "private\_key\_bits" => 1024, "private\_key\_type" => OPENSSL\_KEYTYPE\_RSA,);



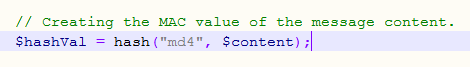
**Figure 2:** Private Key configuration.

As the message digest algorithm of the private key “SHA512” has been used. This is a type of hash function and mainly used for salting values. During the research done, it was revealed that “SHA1” algorithm has some security vulnerabilities. Therefore, SHA512 which is a function of SHA 2 category has been used.

The second most important part of the configuration is the key bits. In this proposed system we have created a private key with 1024 bits, which is considered to be harder to crack than a 512 or 256 bit key. Bits higher than 4096 has not being used due to the fact that it consumes greater processing time and additional over head in data transmission. Along with that, the type of the key has been chosen to be RSA, as opposed to DSA2, considering its added advantages in generating a much secure and, hard to crack keys. Figure 2, depicts the configuration array and the key generation mechanism.

**Generating the MAC value**

After creating the key pair, certificate signing request (CSR) was created by appending the public key, with the web site owner’s information. Then, the message authentication code (MAC) was generated by applying the hash function to the combined massage, which was later encrypted using the site owner’s private key. When sending the CSR to the CA, MAC value will also be appended to it, in order to make it possible or the CA to verify that the request has been received from the valid party (to overcome repudiation) and, the message was not tampered while transmission.



**Figure 3:** Calculating the MAC

As depicted in figure 3, the hash function used for obtaining the hash value of the message is “md4”. This was used mainly considering the processing time and complexity of SHA category functions (e.g.: SHA512).

**Verification of the Request**

On the CA’s end, this digital fingerprint will be decrypted using the public key of the site owner, which has been sent through the request. Then this original hash value would be compared over the second calculation of the received message, to verify the identity of the sender.

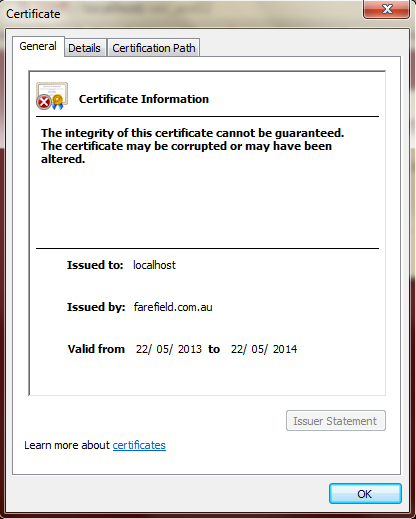


**Figure 4:** Public key decryption.

Figure 4 depicts the functionality to decrypt the digital fingerprint using the public key.

After the verification of the CSR, CA will create the certificate with information such as issued to, issued by, valid from and valid to dates. Along with that, the CA public key will also be appended to the message. Finally, the hash of the created message will be generated and, that will be encrypted with CA’s private key, as the CA’s digital fingerprint. The similar process will be followed by the site owner, to verify the certificated was received from the intended CA.

According to the proposed system, the certificated can be viewed by the clients of the web site, on a web page itself, as well as using the browser address bar notification, since the certificates have been deployed in the server, by the web site administrator. Figure 5, illustrates the certificated which will be displayed by the browsers.



**Figure 5:** Certificate displayed by the browser.

**Reference**

[1] <http://technet.microsoft.com/en-us/library/bb123848(v=exchg.65).aspx>

[2]<http://people.csail.mit.edu/rivest/pubs/Riv98b.prepub.pdf>

[3][http://link.springer.com/chapter/10.1007/978-3-540-24632-9\_28#page-1](http://link.springer.com/chapter/10.1007/978-3-540-24632-9_28" \l "page-1) 2004, Page 1 Introduction

[4]<http://en.wikipedia.org/wiki/Public_key_certificate>.

[5]<http://www.cis.strath.ac.uk/cis/research/publications/papers/strath_cis_publication_2275.pdf>

[6] [http://www.cwi.nl/system/files/PhD-Thesis-Marc-Stevens-Attacks-on-Hash-Functions-and-Applications.pdf 2012](http://www.cwi.nl/system/files/PhD-Thesis-Marc-Stevens-Attacks-on-Hash-Functions-and-Applications.pdf%20%202012), Page 5

[7] [http://www.terena.org/activities/campus-bp/pdf/gn3-na3-t4-abpd106.pdf 2011](http://www.terena.org/activities/campus-bp/pdf/gn3-na3-t4-abpd106.pdf%20%202011), Page 14

[8]<http://paper.ijcsns.org/07_book/201206/20120605.pdf> 2012, Page 1

[9]<http://en.wikipedia.org/wiki/Revocation_list>.

[10] <http://www.terena.org/activities/campus-bp/pdf/gn3-na3-t4-abpd106.pdf> 2011, Page 18

[11] <http://www.hjp.at/doc/rfc/rfc2560.html> 1999, Page 2

[12]<https://www.google.com.au/url?sa=t&rct=j&q=&esrc=s&source=web&cd=13&ved=0CDgQFjACOAo&url=http%3A%2F%2Fwww.jmeds.eu%2Findex.php%2Fjmeds%2Farticle%2Fdownload%2FPKI-Interoperability-Based-on-Online-Certificate-Validation%2F33&ei=AmWcUYeXLMSHiQeC74DQDA&usg=AFQjCNGF8qqTs1k9qhRco-emnn2hP8AH-A&bvm=bv.46751780,d.aGc&cad=rja>

2011, Page 18, part 1.1