Web Content Security System Based on Member's Password

May Zune Zaw, Thwe Mu Han
Computer University, Mandalay
babypinnk@gmail.com

Abstract

A higher degree of security and control is required to protect the content of the system from unauthorized copying, sharing, modifying, etc. In order to lock the information to individual computers and to be unable to share with others, passwords have to be managed while distributing the content over the web. A Secure Hashing Algorithm can be used to protect the user name, password and content of the message by changing it to message digest. A message digest can be signed by the help of Digital Signature Scheme which allows to "signing" documents and this can verify the validity of authentic signatures.

This system is intended to develop user validity checking system to be secure user password and its web site, serving as a message passing system between registered members. It is a system to manage security of a confidential system using Secure Hashing Algorithm (SHA-1) with strong 2^{64} bit encryption mechanism and Digital Signature.

1. Introduction

A web service is a piece of business logic, located somewhere on the Internet, that is accessible through standard-based Internet protocols. Using a web service could be as simple as logging into a site or as complex as facilitating a multi-organization business negotiation. Conventional password schemes involve time-invariant passwords, which provide so-called weak authentication.

Authentication is a service related to identification. Authentication is required to limit access to resources, to identify participants in transactions, and to create seamless personalization of information based on identity [9].

The Web services security dimensions have been defined as: secure messaging, resource protection, negotiation of contracts, trust management, and security properties [9].

Authorizations for Web services are often done through custom implementations such as Hashing and Signing by Secure Hashing Algorithms and Digital Signature Scheme [9].

The remainder of this paper is organized as follows. Section 2 includes detailed discussion about related works. Discussion of Secure Hash Function is described in section 3. Digital Signatures are explained in section 4. System design and implementation are expressed in section 5. The last and final section is concluded about the system.

2. Related Works

Digital Signature is implemented as a hash function that maps large space of all possible messages to a smaller space of hash values. By encoding the basic message, sender does not ensure its integrity, even if the key has not been compromised. The technique that protects the data integrity is based on a one-way hash function *h* that maps a random length text into a fixed size array of bits [9].

An encryption method is presented with the novel property that publicly revealing an encryption key does not thereby reveal the corresponding decryption key. As a consequence, a message can be "signed" using a privately held decryption key. Anyone can verify this signature using the corresponding publicly revealed encryption key. Signatures cannot be forged, and a signer cannot later deny the validity of his signature. This has obvious applications in "electronic mail" and "electronic funds transfer" systems [10].

The importance of E-commerce, electronic data exchange, software distribution and wireless communication has resulted in the necessity of data integrity of which digital signature is playing the main role. The SHA-1 architecture addressed in this paper is to speed up the Digital signature process by means of hardware realization. The design of the SHA-1 computing module is fairly simple. The Secure Hash Algorithm (SHA-1) required in the

Digital Signature Algorithm (DSA) was implemented [11].

3. Secure Hash Function

A hash function is a computationally efficient function mapping binary strings of arbitrary length to binary strings of some fixed length, called hash-value.

3.1 SHA-1 Hashing Algorithm

The hash function SHA-1 takes a message of length less than 2^{64} bits and produces a 160-bit hash value.

The input message is padded and then processed in 512-bit blocks in the Damgard/Merkle iterative structure. Each iterations invokes a so-called compression function which takes a 160-bit chaining value and a 512-bit message block and outputs another 160-bit chaining value. The initial chaining value (called IV) is a set of fixed constants, and the final chaining value is the hash of the message.

The initial chaining value $IV = (a_0, b_0, c_0, d_0, e_0)$ is defined as: (0*x*67452301, 0xefcdab89, 0x98badcfe, 0x10325476, 0xc3d2e1f0)

Each round employs a different Boolean function f_i and constant k_i.

For SHA-1, the initial hash value, $H^{(0)}$, shall consist of the following five 32-bit words, in hex:

$H_0^{(0)}$ = 67452301
$H_1^{(0)}$ = efcdab89
$H_2^{(0)}$ = 98badcfe
$H_3^{(0)}$ = 10325476
$H_4^{(0)}$ = c3d2e1f0**.**

SHA-1 may be used to hash a message, *M*, having a length of λ bits, where $0 \le \lambda < 2^{64}$. The final result of SHA-1 is a 160-bit message digest [1].

SHA-1 Hash Computation: The SHA-1 hash computation uses functions and constants previously defined. Addition (+) is performed modulo 2^{32}.

After preprocessing is completed, each message block, $M^{(1)}$, $M^{(2)}$,…, $M^{(N)}$, is processed in order, using the following steps:

For $i = 1$ to N:

{

1. Prepare the message schedule, $\{W_t\}$:

$$W_t = \begin{cases} M_t^{(i)} & 0 \le t \le 15 \\ ROTL^1(W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) & 16 \le t \le 79 \end{cases}$$

2. Initialize the five working variables, ***a***, ***b***, ***c***, ***d***, and ***e***, with the (i-1) st hash value:

a= $H_0^{(i-1)}$
b= $H_1^{(i-1)}$
c= $H_2^{(i-1)}$
d= $H_3^{(i-1)}$
e= $H_4^{(i-1)}$

3. For t = 0 to 79:

{
T= $ROTL^5(a) + f_t(b, c, d) + e + K_t + W_t$
e= d
d= c
c= $ROTL^{30}(b)$
b= a
a= T
}

4. Compute the i^{th} intermediate hash value $H^{(i)}$:

{
$H_0^{(i)} = a + H_0^{(i-1)}$
$H_1^{(i)} = b + H_1^{(i-1)}$
$H_2^{(i)} = c + H_2^{(i-1)}$
$H_3^{(i)} = d + H_3^{(i-1)}$
$H_4^{(i)} = e + H_4^{(i-1)}$
}

}

After repeating steps one through four a total of N times (i.e., after processing $M^{(N)}$), the resulting 160-bit message digest of the message, *M*, is

$H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)}$.

4. Digital Signature

A cryptographic primitive which is fundamental in authentication, authorization, and non-repudiation is the digital signature. The purpose of a Digital Signature is to provide a means for an entity to bind its identity to a piece of information. The process of signing entails transforming the message and some secret information held by the entity into a tag called a signature. Digital Signature Scheme is a type of asymmetric cryptography used to simulate the security properties of a handwritten signature on the paper. Digital Signature Schemes consist of at least three algorithms: (1) a key generation algorithm, (2) a signature algorithm, and (3) a verification algorithm.

4.1 Key Generation Algorithm for the Digital Signature:

Each entity creates a public key and corresponding private key.

Each entity A should do the following:

1. Select a prime number q such that $2^{159} < q < 2^{160}$.
2. Choose t so that $0 \le t \le 8$, and select a prime number p where $2^{511+64t} < p < 2^{512+64t}$, with the property that q divides (p − 1).

3. (Select a generator α of the unique cyclic group of order q in Z^*_p.)
   3.1 Select an element $g \in Z^*_p$
   And compute $\alpha = g^{(p-1)/q} \bmod p$.
   3.2 If $\alpha = 1$ then go to step 3.1.
4. Select a random integer 'a' such that $1 \le a \le q - 1$.
5. Compute $y = \alpha^a \bmod p$.
6. A's public key is (p, q, α, y); A's private key is a.

Note (*generation of primes* p *and* q) In Key Generation Algorithm it must select the prime q first and then try to find a prime p such that q divides (p−1) [2].

4.2 Algorithm: Digital Signature Generation and Verification:

Entity A signs a binary message m of arbitrary length. Any entity B can verify this signature by using A's public key.

1. Signature Generation.

Entity A should do the following:
(a) Select a random secret integer k, $0 < k < q$.
(b) Compute $r = (\alpha^k \bmod p) \bmod q$.
(c) Compute $k^{-1} \bmod q$.
(d) Compute $s = k^{-1} \{h(m) + a\, r\} \bmod q$.
(e) A's signature for m is the pair (r, s).

2. Verification.

To verify A's signature (r, s) on m, B should do the following:
(a) Obtain A's authentic public key (p, q, α, y).
(b) Verify that $0 < r < q$ and $0 < s < q$; if not, Then reject the signature.
(c) Compute $w = s^{-1} \bmod q$ and h (m).
(d) Compute $u_1 = w \times h(m) \bmod q$ and $u_2 = r\, w \bmod q$.
(e) Compute $v = (\alpha^{u1} y^{u2} \bmod p) \bmod q$.
(f) Accept the signature if and only if v = r.

5. System Design and Implementation

Anyone who wants to access this system must register first as the system will only permit the registered user to access. If registered user enters the system, the system will accept user name and password and then check valid user. The secure hash algorithm SHA-1 is applied to produce message digest and then it will be input into the Digital Signature Scheme. The signed message is decrypted and verified. If valid, user can utilize the web site. In registering process, system changed the user name and password to message digest and then it must be signed with the Digital Signature Scheme. Signed message digest and original user name and password will be saved. The system can verify the signature and check whether the name and password is changed or not.

In checking user name and password, original name and password is changed to message digest by using SHA-1 algorithm, and compare it with login message(digest), if they are the same, there is no change and permit to use the website as shown in Figure 1.

The system is implemented by using C#.Net programming language. The first and main page of the system consists of four menus: Home, Latest, Login, and Registration.

After member login, members can send messages to other members who are online. In message sending process, firstly choose member to send message, and secondly type message and then thirdly hash it by SHA-1 algorithm, and fourthly sign it with Digital Signature and finally send it to choosing member as in Figure2.

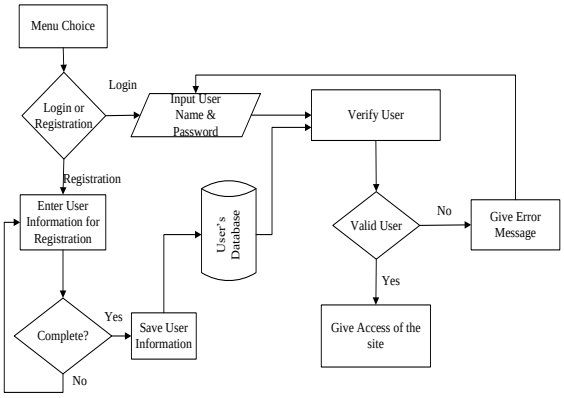
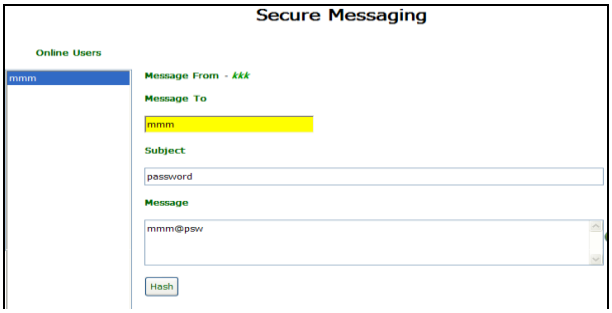


Figure1. System Flow Diagram



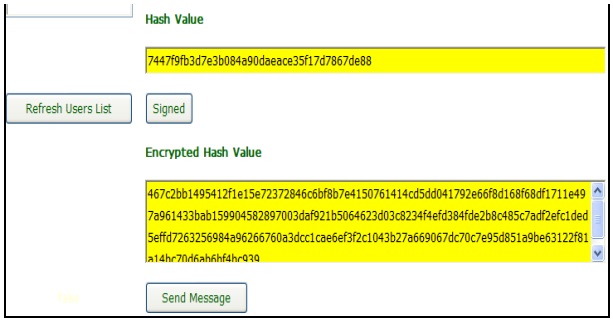


Figure2. User Input, Hash Value and Signed Message

In receiver page, the receiver can select, delete and verify the message. Encrypted hash value can see, shown in Figure4.

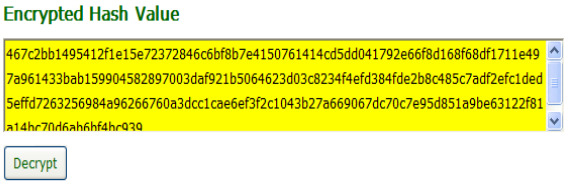


Figure4. Encrypted Hash Value

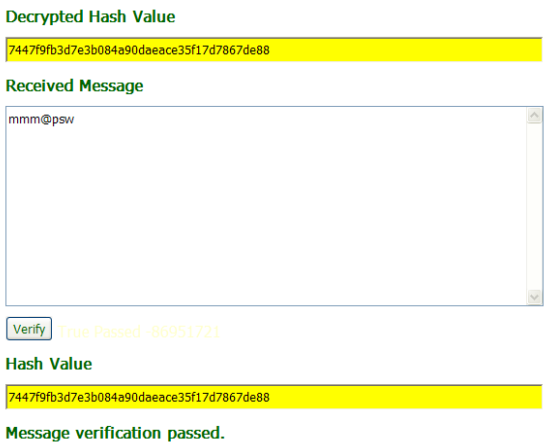


Figure 5. Decrypted Hash Value and Received Message

If the receivers decrypt the encrypted hash value, click the "Decrypt" button. If the original message is not altered, the decrypted hash value and verified hash value is same. Message verification is passed. If the original message is altered, the verified hash value is changed. Message verification is failed as shown in Figure 5.

6. Conclusion

This paper has investigated the cryptographic method for data security during communication. The SHA-1 secure hashing algorithm is used to sign a message of arbitrary block sizes, to change the message digest of fixed size length. The message digest produced by SHA-1 is then input to the Digital Signature Scheme and then it signs on the message digest. The basic idea is: when the user wants to use the web site, user must input user name and password for the system to check it by SHA-1 and Digital Signature. The proposed system can be securing the web content by controlling and checking the name and password. To be seen clearly the validity checking of user name and password by using SHA-1and Digital Signature, the proposed system is included the secure messaging facility. Message sending illustrate firstly the system accept user name and password and secondly then change to message digest and sign this message digest by Digital Signature Scheme. To illustrate the verification process, in the receiver page the signature by using verification algorithm and check the user name and password by the same hashing algorithm. It can be known if it is valid or not. This system is intended to monitor an accidental or an intentional change to the data. If the hash value is changed, system can compare two digests and provide user with error message.

7. References

[1] "Federal Information Processing Standards" Publication 180-2, 2002 August 1, Announcing the SECURE HASH STANDARD

[2] A. Menezes, P. van Oorschot, and S. Vanstone, "Handbook of Applied Cryptography", CRC Press, 1996.

[3] W.Xiaoyun, Y.Lisa Yin, H.Yu, "Finding Collisions in the Full SHA-1", Shandong University, Jinan 250100.

[4] I. M. Khamitov, Alkorsoft & B.Tavrichesky, A.G. Moshonkin, Alkorsoft, A. L. Smirnov, "Blind Unanticipated RSA-Signature Schemes", Alkorsoft & Steklov Mathematical Institute, smirnov@paycash.ru.

[5] Wikipedia, SHA_hash_functions.

[6] Foundations of Cryptography, Lecture 5: Signatures and pseudo-random generators.

[7] Mathematics of Public Key Cryptography, Steven Galbraith, http://www.isg.rhul.ac.uk/~sdg/crypto-book/.

[8] Cryptography and network security.

[9] M.Tuba, N.Stanarevic, P.Strbac, J.Novakovic, Novi Sad J. Math, "Impact of Hash Function Non-Uniformity on Digital Signature Security", Vol. 38, No. 3, 2008, 201-208.

[10]R.L. Rivest, A. Shamir, and L. Adleman_"A Method for Obtaining Digital Signatures and Public-Key Cryptosystems"

[11] S. Pongyupinpanich and S. ChoomchuayAn "Architecture for a SHA-1 Applied for DSA"