IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 9, Issue 6 (Mar. - Apr. 2013), PP 28-33
www.iosrjournals.org

A Study On Comparison Of Algorithm For One Time Password System

K.Divya[1], S.Kalaiarasi[2]
[1] *student in Vels University, M.E Computer Science, Chennai*
[2] *Asst. Professor in Vels University, M.E. Computer Science, Chennai*

Abstract: *People nowadays rely heavily on the Internet since conventional activities or collaborations can be achieved with network services (e.g., web service). Widely deployed web services facilitate and enrich several applications, e.g., online banking, e-commerce, social networks, and cloud computing. user authentication is only handled by text passwords for most websites. Applying text passwords has several critical disadvantages. Safety is a major focus of awareness for operators and users of the website and its many applications, among the difficult problems still inefficiently addressed is identity authentication for purposes of associating a particular user with particular services and authorizations. A request is a way to classify users such that forging recommendation is difficult for adversaries, while providing strong authentication of their chosen identifiers remains easy and convenient for users. The objective of the proposed system is to make online transaction more efficient to the user who uses the website and shops online. This will have a positive impact on user profitability. To make on-line shopping even simpler and safer, a secure processing system is being introduced.*
Index Terms— *authentication, online shopping, text password, web services*

I. Introduction

The text password has been adopted as the primary mean of user authentication for websites. People select their username and text passwords when registering accounts on a website

Thus, most users would choose easy-to-remember passwords (i.e., weak passwords) even if they know the passwords might be unsafe. In researchers have investigated a variety of technology to reduce the negative influence of human factors in the user authentication procedure. Since humans are more adept in remembering graphical passwords than text passwords. we design a user authentication protocol named oPass which leverages a user's cellphone and short message service to thwart password stealing and password reuse attacks. oPass only requires each participating website possesses a unique phone number, and involves a telecommunication service provider in registration and recovery phases. Through oPass, users only need to remember a long-term password for login on all websites. After evaluating the oPass prototype, we believe oPass is efficient and affordable compared with the conventional web authentication mechanisms.Today security concerns are on the rise in all areas such as banks,governmental applications, healthcare industry, military organization,educational institutions, etc. Government organizations are setting standards, passing laws and forcing organizations and agencies to comply with these standards with non-compliance being met with wide-ranging consequences. There are several issues when it comes to security concerns in these numerous and varying industries with one common weak link being passwords. Most systems today rely on static passwords to verify the user's identity. However, such passwords come with major management security concerns. Users tend to use easy-to-guess passwords, use the same password in multiple accounts, write the passwords or store them on their machines, etc. Furthermore, hackers have the option of using many techniques to steal passwords such as shoulder surfing, snooping, sniffing, guessing, etc.

Several proper strategies for using passwords have been proposed. But they didn't meet the company's security concerns. Two factor authentication using devices such as tokens and ATM cards has been proposed to solve the password problem and have shown to be difficult to hack. Two factor authentication is a mechanism which implements two factors and is therefore considered stronger and more secure than the traditionally implemented one factor authentication system. Withdrawing money from an ATM machine utilizes two factor authentications; the user must possess the ATM card, i.e. what you have, and must know a unique personal identification number (PIN), i.e. what you know.Secret word is the most popular form of user authentication on websites due to its convenience and simplicity. The conventional verification table approach has significant drawbacks. Recently, neural networks have been used for password authentication to overcome the shortcomings of traditional approaches. In neural network approaches to password authentication, no verification table is needed; rather, encrypted neural network, main concept of oPass is free users from having to remember or type any passwords into conventional computers for authentication. Existing layered neural network techniques have their limitations such as long training time and recall approximation. In comparison to

existing layered neural network techniques, the proposed method provides better accuracy and quicker response time to registration and password changes.

II. One Time Password

A one-time password (OTP) is a password that is valid for only one login session or transaction. OTPs avoid a number of shortcomings that are associated with traditional (static) passwords. The most important shortcoming that is addressed by OTPs is that, in contrast to static passwords, they are not vulnerable to replay attacks. This means that a potential intruder who manages to record an OTP that was already used to log into a service or to conduct a transaction will not be able to abuse it, since it will be no longer valid. On the downside, OTPs are difficult for human beings to memorize. Therefore they require additional technology to work.OTP generation also typically make use of pseudo or randomness. This is necessary because otherwise it would be easy to predict future OTPs by observing previous ones. Concrete OTP algorithms vary greatly in their details. Various approaches for the generation of OTPs are listed below:

- Based on time-synchronization between the authentication server and the client providing the password (OTPs are valid only for a short period of time)
- Using a mathematical algorithm to generate a new password based on the previous password (OTPs are effectively a chain and must be used in a predefined order).
- Using a mathematical algorithm where the new password is based on a challenge (e.g., a random number chosen by the authentication server or transaction details) and/or a counter.

There are also different ways to make the user aware of the next OTP to use. Some systems use special electronic security tokens that the user carries and that generate OTPs and show them using a small display. Other systems consist of software that runs on the user's mobile phone. Yet other systems generate OTPs on the server-side and send them to the user using an out-of-band channel such as SMS messaging. Finally, in some systems, OTPs are printed on paper that the user is required to carry.

2.1 Methods of delivering OTP

2.1.1Text messaging

A common technology used for the delivery of OTPs is text messaging. Because text messaging is a ubiquitous communication channel, being directly available in nearly all mobile handsets and, through text-to-speech conversion, to any mobile or landline telephone, text messaging has a great potential to reach all consumers with a low total cost to implement. However, the cost of text messaging for each OTP may not be acceptable to some users. OTP over text messaging may be encrypted using an A5/x standard, which several hacking groups report can be successfully decrypted within minutes or seconds,or the OTP over SMS might not be encrypted by one's service-provider at all. In addition to threats from hackers, the mobile phone operator becomes part of the trust chain. In the case of roaming, more than a single mobile phone operator has to be trusted. Anyone using this information may mount a man-in-the-middle attack.Recently Google has started offering OTP to mobile and landline phones for all Google accounts. The user can receive the OTP either as a text message or via an automated call using text-to-speech conversion. In case none of the user's registered phones is accessible, the user can even use one of a set of (up to 10) previously generated one-time backup codes as a secondary authorization factor in place of the dynamically generated OTP, after signing in with their account password.

2.1.2Mobile phones

A mobile phone keeps costs low because a large customer-base already owns a mobile phone for purposes other than generating OTPs. The computing power and storage required for OTPs is usually insignificant compared to that which modern camera-phones and smartphones typically use. Mobile phones additionally support any number of tokens within one installation of the application, allowing a user the ability to authenticate to multiple resources from one device. This solution also provides model-specific applications to the user's mobile phone. However, a cellphone used as a token can be lost, damaged, or stolen.

III. Authentication

Authentication is the process of determining whether someone or something is, in fact, who or what it is declared to be. In private and public computer networks (including the Internet), authentication is commonly done through the use of logon passwords. Knowledge of the password is assumed to guarantee that the user is authentic. Each user registers initially (or is registered by someone else), using an assigned or self-declared password. On each subsequent use, the user must know and use the previously declared password. The weakness in this system for transactions that are significant (such as the exchange of money) is that passwords can often be stolen, accidentally revealed, or forgotten. For this reason, Internet business and many other transactions require a more stringent authentication process. The use of digital certificates issued and verified by

a Certificate Authority (CA) as part of a public key infrastructure is considered likely to become the standard way to perform authentication on the Internet. Three-factor authentication rather than password-based authentication to provide more reliable user authentication. Three-factor authentication depends on what you know (e.g., password), what you have (e.g., token), and who you are (e.g., biometric). To pass the authentication, the user must input a password and provide a pass code generated by the token (e.g., RSA SecureID , and scan her biometric features (e.g., fingerprint or pupil). Three-factor authentication is a comprehensive defense mechanism a against password stealing attacks, but it requires comparative high cost.
Thus, two-factor authentication is more attractive and practical than three-factor authentication. Although many banks support two-factor authentication, it still suffers from the negative influence of human factors, such as the password reuse attack. Users have to memorize another four-digit PIN code to work together with the token, for example RSA SecureID. In addition, users easily forget to bring the token.

IV. Implementation

4.1 Registration phase

The aim of this phase is to allow a user and a server to negotiate a shared secret to authenticate succeeding logins for this user. The user begins by opening the oPass program installed on her cell phone she enters IDu (account id she prefers) and IDs (usually the website url or domain name) to the program. The mobile program sends account id and url to the telecommunication service provider (TSP) through a 3G connection to make a request of registration. Once the TSP received the account id and the url, it can trace the user's phone number based on user's SIMcard. The TSP also plays the role of third-party to distribute a shared key between the user and the server. The shared key is used to encrypt the registration SMS with AES-CBC. The TSP and the server will establish an SSL tunnel to protect the communication. Then the TSP forwards account id, and to the assigned server. Server will generate the corresponding information for this account and reply a response, including server's identity ID, a random seed, and server's phone number. The TSP then forwards id, and a shared key to the user's cell phone. Once reception of the response is finished, the user continues to setup a long-term password with her cell phone.

4.2 Login phase

The login phase begins when the user sends a request to the server through an untrusted browser (on a kiosk). The user uses her cell phone to produce a one-time password, e.g., , and deliver necessary information encrypted with to server via an SMS message. Based on pre shared secret credential, server can verify and authenticate user . The detail flows of the login phase. The protocol starts when user wishes to log into her favourite web server (already registered). However, begins the login procedure by accessing the desired website via a browser on an untrusted kiosk. The browser sends a request to with account IDs. Next, server supplies the ID and a fresh nonce to the browser.Meanwhile, this message is forwarded to the cell phone through GSM Modem. After reception of the message, the cell phone inquiries related information from its database via IDs, which includes server's phone number and other parameters The next step is promoting a dialog for her long-term password. Secret shared credential can regenerate by inputting the correct on the cell phone. The one-time password for current login is recomputed If the received equals the previously generated, the user is legitimate; otherwise, the server will reject this login request. Upon successful verification, the server sends back a success message through the Internet, if the user is successfully log into the server.

4.3 Recovery phase

Recovery phase is designated for some specific conditions; for example, a user may lose her cell phone. The protocol is able to recover oPass setting on her new cell phone assuming she still uses the same phone number (apply a new SIM card with old phone number). Once user installs the oPass program on her new cell phone, she can launch the program to send a recovery request with her account ID and requested server ID to predefined TSP through a 3G connection. As we mentioned before, ID can be the domain name or URL link of server. Similar to registration, TSP can trace her phone number based on her SIM card and forward her account ID and the to server through an SSL tunnel. Once server receives the request, probes the account information in its database to confirm if account is registered or not. If account ID exists, the information used to compute the secret credential will be fetched and be sent back to the user.

This message Procedure of recovery phase includes all necessary elements for generating the next one-time passwords to the user . When the mobile program receives the message, like registration, it forces the user to enter her long-term password to reproduce the correct one-time password. During the last step, the user's cell phone encrypts the secret credential and server nonce to a cipher text. The recovery SMS message is delivered back to the server for checking. Similarly, the server computers and decrypts this message to ensure that user is already recovered. At this point, her new cell phone is recovered and ready to perform further logins. For the next login, one-time password will be used for user authentication.

V. Algorithms

5.1 MD5 algorithm

MD5 algorithm that is used to verify data integrity through the creation of a 128-bit message digest from data input (which may be a message of any length) that is claimed to be as unique to that specific data as a fingerprint is to the specific individual. MD5, which was developed by Professor Ronald L. Rivest of MIT, is intended for use with digital signature applications, which require that large files must be compressed by a secure method before being encrypted with a secret key, under a public key cryptosystem. MD5 is currently a standard, Internet Engineering Task Force (IETF) Request for Comments (RFC) 1321. According to the standard, it is "computationally infeasible" that any two messages that have been input to the MD5 algorithm could have as the output the same message digest, or that a false message could be created through apprehension of the message digest. MD5 is the third message digest algorithm created by Rivest. All three (the others are MD2 and MD4) have similar structures, but MD2 was optimized for 8-bit machines, in comparison with the two later formulas, which are optimized for 32-bit machines. The MD5 algorithm is an extension of MD4, which the critical review found to be fast, but possibly not absolutely secure. In comparison, MD5 is not quite as fast as the MD4 algorithm, but offers much more assurance of data security.

The MD5 widely used cryptographic hash function that produces a 128-bit (16-byte) hash value. Specified in RFC 1321, MD5 has been employed in a wide variety of security applications, and is also commonly used to check data integrity. However, it has been shown that MD5 is not collision resistant; as such, MD5 is not suitable for applications like SSL certificates or digital signatures that rely on this property. An MD5 hash is typically expressed as a 32-digit hexadecimal number.MD5 was designed by Ron Rivest in 1991 to replace an earlier hash function, MD4. In 1996, a flaw was found with the design of MD5. While it was not a clearly fatal weakness, cryptographers began recommending the use of other algorithms, such as SHA-1 (which has since been found also to be vulnerable). In 2004, more serious flaws were discovered, making further use of the algorithm for security purposes questionable; specifically, a group of researchers described how to create a pair of files that share the same MD5 checksum. Further advances were made in breaking MD5 in 2005, 2006, and 2007. In an attack on MD5 published in December 2008, a group of researchers used this technique to fake SSL certificate validity.

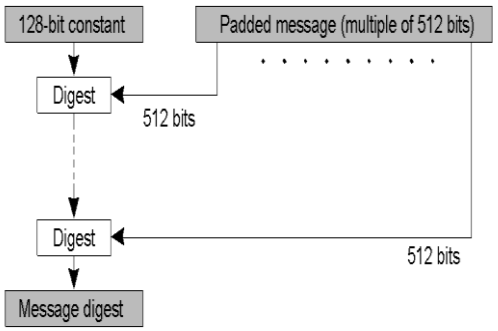

Fig. 1 MD5 algorithm structure

5.1.1 Implementation steps

Step1: Append padding bits

The input message is "padded" (extended) so that its length (in bits) equals to 448 mod 512. Padding is always performed, even if the length of the message is already 448 mod 512.

Padding is performed as follows: a single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits of the padded message becomes congruent to 448 mod 512. At least one bit and at most 512 bits are appended.

Step2. Append length

A 64-bit representation of the length of the message is appended to the result of step1. If the length of the message is greater than 2^64, only the low-order 64 bits will be used.

The resulting message (after padding with bits and with b) has a length that is an exact multiple of 512 bits. The input message will have a length that is an exact multiple of 16 (32-bit) words.

Step3. Initialize MD buffer

A four-word buffer (A, B, C, D) is used to compute the message digest. Each of A, B, C, D is a 32-bit register. These registers are initialized to the following values in hexadecimal, low-order bytes first):

word A: 01 23 45 67

word B: 89 ab cd ef
word C: fe dc ba 98
word D: 76 54 32 10

Step4. Process message in 16-word blocks

Four functions will be defined such that each function takes an input of three 32-bit words and produces a 32-bit word output.

F (X, Y, Z) = XY or not (X) Z
G (X, Y, Z) = XZ or Y not (Z)
H (X, Y, Z) = X xor Y xor Z
I (X, Y, Z) = Y xor (X or not (Z))

5.2 SHA-256 algorithm

The SHA-256 algorithm is very similar in structure to SHA-1, but not only does it use eight, rather than five, 32-bit sub blocks, but there are other ways in which it is not analogous. For SHA-256, the message is padded, and divided into 512-bit blocks, in the same way as for SHA-1. From each block, considered as 16 32-bit words, 64 (rather than 80) 32-bit words are produced, the first 16 being the block itself, and the remaining words being the sum, modulo 2^32, of the following quantities; the word 16 words ago, the word 7 words ago.

The XOR of the following three quantities:
- the word 2 words ago rotated right 17 places
- that word rotated right 19 places
- that word shifted right 10 places;

The XOR of the following three quantities:
- the word 15 words ago rotated right 7 places
- that word rotated right 18 places
- that word shifted right 3 places.

One round of the part of SHA-256 that looks like a round of a block cipher is performed for each of these 64 words. For the first block, the initial input values to SHA-256 are:

6A09E667 BB67AE85 3C6EF372 A54FF53A
510E527F 9B05688C 1F83D9AB 5BE0CD19

which are the beginnings, in hexadecimal, of the fractional parts of the square roots of 2, 3, 5, 7, 11, 13, 17, and 19.

The round function of SHA-256 is as follows:

An intermediate result is calculated, which is equal to the modulo 2^32 sum of

- The XOR of the following three quantities:
  - the fifth word in the block rotated right 6 places
  - that word rotated right 11 places
  - that word rotated right 25 places;
- a word consisting of those bits in the sixth word of the block which correspond to bits of the fifth word of the block that are ones, and those bits in the seventh word of the block that correspond to bits of the fifth word of the block that are zeroes;
- the current one of the 64 words to which the 16 word block is expanded;
- the current one of 64 constants introduced into this phase.

The eighth word of the block is modified by having this intermediate result added to it modulo 2^32.The resulting incompletely modified new value of the eighth word in the block is then added to the fourth word in the block modulo 2^32.Then, two additional quantities are added to the eighth word in the block modulo 2^32:

- A word whose bits are 1 if and only if two of the corresponding three bits taken from each of the first, second, and third words in the block are 1;
- The XOR of the following three quantities:
  - The first word in the block rotated right 2 bits,
  - that word rotated right 13 bits,
  - that word rotated right 22 bits.

Finally, each of the eight words of the block that will ultimately become the hash is moved to the position of the next word in the block, with the first word in the block being replaced by the modified eighth word in the block.The 64 constant words, added to each word in the expanded block, are:

428A2F98 71374491 B5C0FBCF E9B5DBA5 3956C25B 59F111F1 923F82A4 AB1C5ED5
D807AA98 12835B01 243185BE 550C7DC3 72BE5D74 80DEB1FE 9BDC06A7 C19BF174
E49B69C1 EFBE4786 0FC19DC6 240CA1CC 2DE92C6F 4A7484AA 5CB0A9DC 76F988DA
983E5152 A831C66D B00327C8 BF597FC7 C6E00BF3 D5A79147 06CA6351 14292967

27B70A85 2E1B2138 4D2C6DFC 53380D13 650A7354 766A0ABB 81C2C92E 92722C85
A2BFE8A1 A81A664B C24B8B70 C76C51A3 D192E819 D6990624 F40E3585 106AA070
19A4C116 1E376C08 2748774C 34B0BCB5 391C0CB3 4ED8AA4A 5B9CCA4F 682E6FF3
748F82EE 78A5636F 84C87814 8CC70208 90BEFFFA A4506CEB BEF9A3F7 C67178F2
After this has been done 64 times, the final result is the sum, by individual words modulo 2^32, of the result of this transformation and the original eight-word input.Thus, one important difference between SHA-256 and SHA-1 is that the nonlinear functions do not change during the hashing of a block, but instead of having only four constants, each of which is used for 20 words, there are now 64 constants, each used for only one word.

VI. Performance Analysis

The analysis of algorithms is the determination of the number of resources (such as time and storage) necessary to execute them. Most algorithms are designed to work with inputs of arbitrary length. Usually the efficiency or running time of an algorithm is stated as a function relating the input length to the number of steps (time complexity) or storage locations.

Algorithm analysis is an important part of a broader computational complexity theory, which provides theoretical estimates for the resources needed by any algorithm which solves a given computational problem. These estimates provide an insight into reasonable directions of search for efficient algorithms.

Table1. Comparison of algorithms

Algorithm	Ouput size (bits)	Word Size (bits)	Rounds	collisions	Performance (MiB/s)
MD5	128	32	64	Yes	255
SHA-256	256	32	64	None	111

Through the experimental results sha-256 algorithm is better than md5, since collisions are not occurred in sha-256 algorithm. The speed efficiency of md5 is low compared to sha-256.

VII. Conclusion

OTP is to eliminate the negative influence of human factors as much as possible. Through OTP each user only needs to remember a long-term password which has been used to protect the cellphone. Users are free from typing any passwords into untrusted computers for login on all websites. Compared with previous schemes, OTP is the first user authentication protocol to prevent password stealing (i.e., phishing, keylogger, and malware) and password reuse attacks simultaneously. The reason is that OTP adopts the one-time password approach to ensure independence between each login. The OTP is implemented by using two cryptographic hashing algorithm such as MD5 and SHA-256.
Comparing both algorithms SHA-256 performance is efficient than MD5 algorithm.

References

[1]. Hung-Min Sun, Yao-Hsin Chen, and Yue-Hsun Lin, " oPass: A User Authentication Protocol Resistant to Password Stealing and Password Reuse Attacks",in IEEE transactions on information forensics and security, vol. 7, no. 2, april 2012.
[2]. S. Chiasson, A. Forget, E. Stobert, P. C. van Oorschot, and R. Biddle, "Multiple password interference in text passwords and click-based graphical passwords," in CCS '09: Proc. 16th ACM Conf. Computer Communications Security, New York, 2009, pp. 500–511, ACM.
[3]. S. Gawand E. W. Felten, "Password management strategies for online accounts, in SOUPS '06: Proc. 2nd Symp. Usable Privacy Security, New York, 2006,p 44– 55, ACM.
[4]. K.-P. Yee and K. Sitaker, "Passpet: Convenient password management andphishing protection," in SOUPS '06: Proc. 2nd Symp. Usable Privacy Security, New York, 2006, pp. 32–43, ACM.
[5]. M.Wu, S. Garfinkel, and R. Miller, "Secure web authentication with mobile phones," in DIMACS Workshop Usable Privacy Security Software,Citeseer, 2004.
[6]. A. Perrig and D. Song, "Hash visualization: A new technique to improve real-world security," in Proc. Int.Workshop Cryptographic Techniques E-Commerce, Citeseer, 1999, pp. 13.
[7]. Florencio and Cormac Herley, "One-Time Password Access to Any Server without Changing the Server" in Microsoft Research, One Microsoft Way, Redmond, WA.
[8]. Joseph D. Touch, "Performance Analysis of MD5" *in USC / Information Sciences Institute*