Implementation of secure login algorithm over internet using steganography

Sahir Riyaz Khan[1], Tejaswi Ogirala[2], Ravi Kiran Modali[3], Vandana Mishra[4]

[1,4]Computer Science, SRM University
[2.3]Electronics and Communications, SRM University

Abstract-This paper suggests a new scheme where steganography is used to provide a highly secured authentication. The use of character encoding and decoding in an image is used which provides a very high form of security over the web applications. The proposed idea is an alternative to biometrics and RSA tokens. Various attacks pertaining to passwords as brute-force attack, shoulder surfing attack, MIM attack, dictionary attack and password guessing can be prevented. Thus putting a barrier to identity theft, password misuse, loss of confidential information etc., which thereby secures it from hackers.

Index Terms- Steganography, Secured Authentication, Hash, Image ID.

I. INTRODUCTION

According to the “Internet Security Threat report” from Symantec the volume and sophistication of online attacks continues to increase. In fact, the daily volume of Web-based attacks increased by 93% from 2009 to 2010, while attack toolkits grew to account for two-thirds of all Web-based threats. Notably, the report found that Web-based attacks are hitting businesses' bottom lines, due to the cost of data breaches. In particular, the report found that hacking results in an average of 262,767 identities exposed per data breach incident. Accounting for many fewer lost records are insiders (68,418), theft or loss (67,528), insecure policies (30,572), or fraud (6,353).

II. WEB AUTHENTICATION TERMINOLOGIES

The process of proving one's identity. (The primary forms of host-to-host authentication on the Internet today are name-based or address-based, both of which are notoriously weak.)

A. HTTPS

HTTPS is a protocol to transfer encrypted data over the Web. There are two primary differences between an HTTPS and an HTTP connection work.

1) HTTPS connects on port 443, while HTTP is on port 80.
2) HTTPS encrypts the data sent and received with SSL, while HTTP sends it all as plain text.

HTTP is unsecured and is subject to man-in-the-middle and eavesdropping attacks, which can let attackers gain access to website accounts and sensitive information. HTTPS is designed to withstand such attacks and is considered secure against such attacks (with the exception of older deprecated versions of SSL). HTTP sends the data collected over the Internet in plain text. This means that if you have a form asking for a credit card number, that credit card number can be intercepted by anyone with a packet sniffer. Since there are many sniffer software tools, this could be anyone at all. By collecting credit card information over an HTTP (not HTTPS) connection, you are broadcasting that credit card information to the world. And the only way your customer will learn it was stolen is when it's maxed out by a thief. After the HTTPS Certificate your hosting provider will need to set up the certificate in your Web server so that every time a page is accessed via the https:// protocol, it hits the secure server. Once that is set up, you can start building your Web pages that need to be secure.

B. Hash Functions

Hash Functions uses a mathematical transformation to irreversibly "encrypt" information. Hash functions, also called message digests and one-way encryption, are algorithms that, in some sense, use no key. Instead, a fixed-length hash value is computed based upon the plain text that makes it impossible for either the contents or length of the plain text to be recovered. Hash algorithms are typically used to provide a digital fingerprint of a file's contents often used to ensure that the file has not been altered by an intruder or virus. Hash functions are also commonly employed by many operating systems to encrypt passwords. Hash functions, then, provide a measure of the integrity of a file. Hash functions, are well-suited for ensuring data integrity because any change made to the contents of a message will result in the receiver calculating a different hash value than the one placed in the transmission by the sender. Since it is highly unlikely that two different messages will yield the same hash value, data integrity is ensured to a high degree of confidence.

Hash functions are also used to build caches for large data sets stored in slow media. A cache is generally simpler than a hashed search table, since any collision can be resolved by discarding or writing back the older of the two colliding items. Hash functions are an essential ingredient of the Bloom filter, a compact data structure that provides an enclosing approximation to a set of them. The cost of computing a hash function must be small enough to make a hashing-based solution more efficient than alternative approaches.

Hash algorithms that are in common use today include:

1) Message Digest (MD) algorithms: A series of byte-oriented algorithms that produce a 128-bit hash value from an arbitrary-length message.
   - MD2 (RFC 1319): Designed for systems with limited memory, such as smart cards.
   - MD4 (RFC 1320): Developed by Rivest, similar to MD2 but designed specifically for fast processing in software.
   - MD5 (RFC 1321): Also developed by Rivest after potential weaknesses were reported in MD4; this scHash Calc
   - HAVAL (HAsh of VAriable Length)
   - Tigerheme is similar to MD4 but is slower because more manipulation is made to the original data. MD5 has been implemented in a large number of products although several weaknesses in the algorithm were demonstrated by German cryptographer Hans Dobbertin in 1996.

Certain extensions of hash functions are used for a variety of information security and digital forensics applications, such as:

2) Hash libraries are sets of hash values corresponding to known files. A hash library of known good files, for example, might be a set of files known to be a part of an operating system, while a hash library of known bad files might be of a set of known child pornographic images.
3) Rolling hashes refer to a set of hash values that are computed based upon a fixed-length "sliding window" through the input. As an example, a hash value might be computed on bytes 1-10 of a file, then on bytes 2-11, 3-12, 4-13, etc.
4) Fuzzy hashes are an area of intense research and represent hash values that represent two inputs that are similar. Fuzzy hashes are used to detect documents, images, or other files that are close to each other with respect to content.

Hash functions, are well-suited for ensuring data integrity because any change made to the contents of a message will result in the receiver calculating a different hash value than the one placed in the transmission by the sender. Since it is highly unlikely that two different messages will yield the same hash value, data integrity is ensured to a high degree of confidence.

C. SSL

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols that provide communication security over the Internet. TLS and SSL encrypt the segments of network connections above the Transport Layer, using asymmetric cryptography for privacy and a keyed message authentication code for message reliability.

Several versions of the protocols are in widespread use in applications such as web browsing, electronic mail, Internet faxing, instant messaging and voice-over-IP (VoIP).

TLS is an IETF standards track protocol, last updated in RFC 5246 and is based on the earlier SSL specifications developed by Netscape Corporation.

SSL is an Internet security protocol used by Internet browsers and web servers to transmit sensitive information. An SSL certificate serves two essential purposes: distributing the public key and verifying the identity of the server so users know they aren't sending their information to the wrong server. It can only properly verify the identity of the server when it is signed by a trusted third party. A self signed certificate is a certificate that is signed by itself rather than a trusted authority. Since any attacker can create a self signed certificate and launch a man-in-the-middle attack, a user can't know whether they are sending their encrypted information to the server or an attacker. Because of this, you will almost never want to use a self signed certificate on a public Java server that requires anonymous visitors to connect to your site. **S***ecure* **S***ockets* **L***ayer* is a protocol developed by Netscape for transmitting private documents via the Internet. SSL uses a cryptographic system that uses two keys to encrypt data − a public key known to everyone and a private or secret key known only to the recipient of the message. Both Netscape Navigator and Internet Explorer support SSL, and many Web sites use the protocol to obtain confidential user information, such as credit card numbers. By convention, URLs that require an SSL connection start with *https:* instead of *http.*

D. CAPTCHA

CAPTCHA is a type of challenge-response test used in computing as an attempt to ensure that the response is generated by a person. The process usually involves one computer (a server) asking a user to complete a simple test which the computer is able to generate and grade. Because other computers are supposedly unable to solve the CAPTCHA, any user entering a correct solution is presumed to be human. Thus, it is sometimes described as a reverse Turing test, because it is administered by a machine and targeted to a human, in contrast to the standard Turing test that is typically administered by a human and targeted to a machine. A common type of CAPTCHA requires the user to type letters or digits from a distorted image that appears on the screen.

A CAPTCHA is a means of automatically generating challenges which intends to:

1) Provide a problem easy enough for all humans to solve.
2) Prevent standard automated software from filling out a form, unless it is specially designed to circumvent specific CAPTCHA systems.

Some researchers promote image recognition CAPTCHAs as a possible alternative for text-based CAPTCHAs. Computer-based recognition algorithms require the extraction of color, texture, shape, or special point features, which cannot be correctly extracted after the designed distortions. However, humans can still recognize the original concept depicted in the images even with these distortions.

CAPTCHA is vulnerable to a relay attack that uses humans to solve the puzzles. One approach involves relaying the puzzles to a group of human operators who can solve CAPTCHAs. In this scheme, a computer fills out a form and when it reaches a CAPTCHA, it gives the CAPTCHA to the human operator to solve.

III. STEGANOGRAPHY

Steganography is the art and science of writing hidden messages in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message, a form of security through obscurity i.e. messages will appear to be something else: images, articles, shopping lists, or some other cover text and, classically, the hidden message may be in invisible ink between the visible lines of a private letter. Steganography includes the concealment of information within computer files. In digital steganography, electronic communications may include steganographic coding inside of a transport layer, such as a document file, image file, program or protocol. Media files are ideal for steganographic transmission because of their large size. As a simple example, a sender might start with an innocuous image file and adjust the color of every 100th pixel to correspond to a letter in the alphabet, a change so subtle that someone not specifically looking for it is unlikely to notice it.

Modern steganography entered the world in 1985 with the advent of the personal computer being applied to classical steganography problems. Development following that was slow, but has since taken off, going by the number of "stego" programs available: Over 800 digital steganography applications have been identified by the steganography Analysis and Research Center. Digital steganography techniques include:

1) Concealing messages within the lowest bits of noisy images or sound files.
2) Concealing data within encrypted data or within random data. The data to be concealed is first encrypted before being used to overwrite part of a much larger block of encrypted data or a block of random data (an unbreakable cipher like the one-time pad generates cipher texts that look perfectly random if you don't have the private key).
3) Chaffing and winnowing.
4) Mimic functions convert one file to have the statistical profile of another. This can thwart statistical methods that help brute-force attacks identify the right solution in a cipher text-only attack.
5) Concealed messages in tampered executable files, exploiting redundancy in the targeted instruction set.
6) Pictures embedded in video material (optionally played at slower or faster speed).
7) Injecting imperceptible delays to packets sent over the network from the keyboard. Delays in key presses in some applications (telnet or remote desktop software) can mean a delay in packets, and the delays in the packets can be used to encode data.
8) Changing the order of elements in a set.
9) Content-Aware Steganography hides information in the semantics a human user assigns to a datagram. These systems offer security against a non-human adversary/warden.
10) Messages are fractionalized and the (encrypted) pieces are added as comments of orphaned web-logs (or pin boards on social network platforms). In this case the selection of blogs is the symmetric key that sender and recipient are using; the carrier of the hidden message is the whole blogosphere.
11) Modifying the echo of a sound file (Echo Steganography).
12) Secure Steganography for Audio Signals.

Image bit-plane complexity segmentation steganography (i.e., BPCS-Steganography).

IV. THE PROPOSAL

This paper the authors have incorporated two factor authentication strategies i.e. what user has, what user knows. In our case what user has is an image and what user knows is a pin. We can visualize this case by taking an example of an ATM card, i.e. possession of ATM card and correct pin authenticates a user to perform transaction. Same applies here; our image contains user data which uniquely identifies a user. It contains encoded data which is encoded by coalition of SHA-2 and MD5 hashing algorithm. The Pin acts as a normal password which would be mandatory. The validation of both is done in order to authenticate a user. While sign up user enters some random data, random string which could be combination of all characters present over keyboard, now this data will be hashed with some salt and will be encoded into the image by steganographic algorithms mentioned above. The hashing is iterated to make it much more complex. The user needs not to remember the data he gave for his image id; instead he just needs to remember his pin or password. The hashed value which we generate by using our hashing algorithm is again salted and iteratively hashed and is stored in database. Thus while login a user writes his username and chooses his image id with his pin and clicks submits, and then the hashed data is decoded from the image which is again iteratively hashed with a salt and is compared with the stored hash for the data. If each and every credentials match then login is successful. It is implemented over HTTPS. This proposal thus impedes/stops a brute force or a dictionary attack. This image id can be easily kept in any of the memory devices such as flash drives, hard disk drives or even in the web servers. The image id is of no use unless the correct pin is known. Thus making the authentication much secure and reliable.

V. IMPLEMENTATION

A. *Sign up/register:*

The user interface for the sign up is similar to sign up page in any of the web applications. The only addition would be to enter some random data which might be combination of all characters over a keyboard. This String entered by user will be used to generate hash value using MD5 and SHA-2 hybrid algorithm with some salt. Now this Hashed value will be encoded into an image using algorithms mentioned. After successful sign up, the user would be provided with an image id of 5-10 kilo-bytes in size. All the images produced would look similar. The user must keep it safe. The user should also chose a pin or password at the time of sigh up. This Pin along with the image id act as sole identification criteria for a particular user. The Hashed data which is encoded in image id is again literately hashed to Store it into the database.

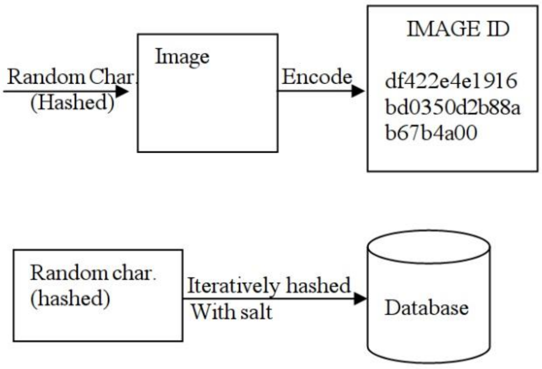

Fig 1: Image created automatically and random data further hashed is stored in the database.

B. *Login*:

At the login time user will be presented a login box where he needs to enter his username, the image id accompanied with the pin. The web application will now check each and every credentials provided by user. The data inside the image id is decoded and again iteratively hashed with salt and is checked against the value stored at database. The pin is also checked against the database as the normal ways. The correct match of all three credentials i.e., username, image id, pin authenticates a user.

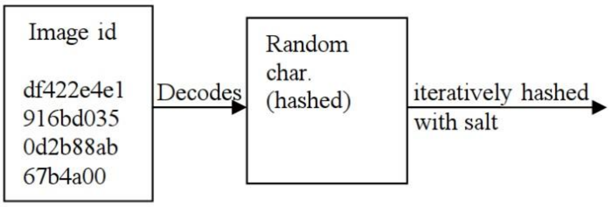

Fig 2: Image id decoded during authentication.

C. *Applicability*:

The above proposed authentication could be used anywhere where user authentication matters a lot. Sites which require having safe user authentication like the corporate sites, banking sites and even federal government sites. This scheme would impede a brute force attack, dictionary attack and password guessing attack. As the user will not have anything to guess for an image id thus it stops password guessing attack. Thus making its authentication secure.

VI. CONCLUSION

The authentication methods proposed above would eliminate the problems with several web attacks as brute-force attack, dictionary attack, and password guessing which would make the web applications immune from various hackers thus making web authentication much secure.

ACKNOWLEDGMENT

The authors wish to acknowledge their family members for their tremendous support and also the role of the faculty of SRM University, NCR Campus, Ghaziabad. This document could not have been completed without their relentless support.

REFERENCES

[1]. Peter Eckersley: Encrypt the Web with the HTTPS Everywhere Firefox Extension EFF blog, 17 June 2010
[2]. "Free SSL Certificates from a Free Certificate Authority". sslshopper.com. Retrieved 2009-10-24.
[3]. Justin Fielding (2007-07-16). "Secure Outlook Web Access with (free) SSL: Part 1". TechRepublic. Retrieved 2009-10-24.
[4]. "SSL Certificate Services". Go Daddy. Retrieved 6 May 2009.
[5]. "Secure Site Pro with EV". VeriSign. Retrieved 6 May 2009.
[6]. Knuth, Donald (1973). The Art of Computer Programming, volume 3, Sorting and Searching. pp. 506–542.
[7]. "Robust Audio Hashing for Content Identification by Jaap Haitsma, Ton Kalker and Job Oostveen"
[8]. Z. Broder. Some applications of Rabin's fingerprinting method. In Sequences II: Methods in Communications, Security, and Computer Science, pp. 143–152. Springer-Verlag, 1993
[9]. Bret Mulvey, Evaluation of CRC32 for Hash Tables, in Hash Functions. Accessed April 10, 2009.
[10]. Bret Mulvey, Evaluation of SHA-1 for Hash Tables, in Hash Functions. Accessed April 10, 2009.
[11]. Thomas Y. C. Woo, Raghuram Bindignavle, Shaowen Su andSimon S. Lam, SNP: An interface for secure network programmingProceedings USENIX Summer Technical Conference, June 1994
[12]. Dierks, T. and E. Rescorla (April 2006). "The Transport Layer Security (TLS) Protocol Version 1.1, RFC 4346".

[13]. National Institute of Standards and Technology (December 2010). "Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program".

[14]. Eric Rescorla (2009-11-05). "Understanding the TLS Renegotiation Attack". Educated Guesswork. Retrieved 2009-11-27.

[15]. McMillan, Robert (2009-11-20). "Security Pro Says New SSL Attack Can Hit Many Sites". PC World. Retrieved 2009-11-27.

[16]. "SSL_CTX_set_options SECURE_RENEGOTIATION".OpenSSL Docs. 2010-02-25. Retrieved 2010-11-18.

[17]. "METHOD AND SYSTEM FOR DISCRIMINATING A HUMAN ACTION FROM A COMPUTERIZED ACTION". 1998. Retrieved 1998-12-31.

[18]. "Latest Status of CAPTCHA Trademark Application". USPTO. 2008-04-21. Retrieved 2008-12-21.

[19]. Amrinder Arora (2007). "Statistics Hacking — Exploiting Vulnerabilities in News Websites" (PDF). International Journal of Computer Science and Network Security 7: 342–347.

[20]. "Breaking CAPTCHAs Without Using OCR". Howard Yeend (pureMango.co.uk). 2005. Retrieved 2006-08-22.

[21]. Pahati, OJ (2001-11-29). "Confounding Carnivore: How to Protect Your Online Privacy". AlterNet. Archived from the original on 2007-07-16. Retrieved 2008-09-02.

[22]. Petitcolas, FAP; Anderson RJ; Kuhn MG (1999). "Information Hiding: A survey" (pdf). Proceedings of the IEEE (special issue)87 (7): 1062–78. doi:10.1109/5.771065. Retrieved 2008-09-02.

[23]. SARC - Digital Steganography Database Exceeds 800 Applications

[24]. Krzysztof Szczypiorski (4 November 2003). "Steganography in TCP/IP Networks. State of the Art and a Proposal of a New System - HICCUPS". Institute of Telecommunications Seminar. Retrieved 17 June 2010.