Crypto-Cloud Integration of oPass

Shivam Aditya[1], Varun Mittal[2]

[1]VIT University, sa.shivam.aditya@gmail.com

[2]VIT University, varun.mittal2011@vit.ac.in

Abstract: One of the most popular forms of user authentication is the Text Passwords. It is due to its convenience and simplicity. Still, the passwords are susceptible to be taken and compromised under various threats and weaknesses. In order to overcome these problems, a protocol called oPass was proposed. A cryptanalysis of it was done. We found out four kinds of attacks which could be done on it i.e. Use of SMS service, Attacks on oPass communication links, Unauthorized intruder access using the master password, Network attacks on untrusted web browser. One of them was Impersonation of the User. In order to overcome these problems in cloud environment, a protocol is proposed based on oPass to implement crypto-cloud integration with oPass which can handle this kind of attack.

***Keywords**:* Cloud, Impersonation, oPass, SMS, Digital Signature, Network Security

1. INTRODUCTION

oPass scheme (Sun, 2012) is a protocol resistant to password stealing and password reuse attacks. In their user authentication protocol named oPass, they leveraged a user's cell phone and short message service (SMS) to avert password stealing and password reuse attacks. The only requirement of oPass is that each participating website possesses a unique phone number and involves a telecommunication service provider (TSP) in registration, login and recovery phases. In oPass protocol, users only need to remember a long-term password for login on all websites. It claimed that it satisfied all the requirements of the user and is immune to various attacks.

It has three phases in it: (1) Registration Phase: The aim of this phase is to register the user in the service for the first time and allow him/her and a server to negotiate a shared secret in the form of OTP (One Time Password) to authenticate succeeding logins for this user. (2) Login Phase: The login phase begins when the user sends a request to the server through an untrusted browser by using a kiosk. It performs all the tasks needed for the proper login and authentication of the user. (3) Recovery phase is a phase only designated for some specific conditions; for an example, a user may lose his/her cell phone. This protocol is able to recover oPass configuration on his/her new cell phone assuming he/she still uses the same phone number by applying for a fresh SIM card with the same phone number given in oPass.

Cloud computing is a type of model for supporting easy on-demand access to a shared pool of configurable computing resources (e.g., servers, networks, applications, storage and services) that can be rapidly purveyed and released with minimal effort or service provider interaction.

Any user having access to the server can utilise its processing power, store data, or use it in various other purposes. This has given the free will to the user to access the application from anywhere in the world instead of using a personal computer every time to execute a native application.

The major features of cloud computing include: (1) On demand self-service (2) Broad network access (3) Resource pooling (4) Rapid elasticity (5) Metered usage. (Ali, 2009)

It's new services include:

(1) Software as a Service (SaaS) - In this applications (word processor, CRM, etc.) or application services (schedule, calendar, etc.) execute in the "cloud" using the interconnectivity of the internet to propagate data.

(2) Platform as a Service (PaaS) - Applications are built in the "cloud" on the platform using a variety of technologies. Development, testing, and production environments (servers, storage, bandwidth, etc.) are billed monthly like hosting. Pay-as-you-go model.

(3) Infrastructure as a Service (IaaS) – In this computation resources (memory, bandwidth, storage, processors etc.) are provided in an as per needed, pay-as-you-go model. It is highly scalable and able to provide from single server up to entire data centers. It can be used to create new opportunities such as Cloud bursting: shifting usage spike traffic to alternate resources.

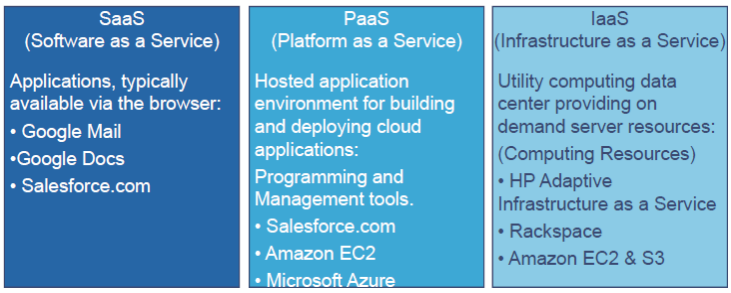

Figure 1. New services of Cloud Computing

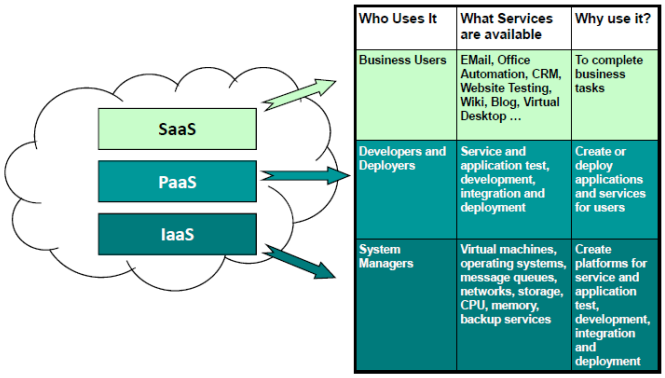

Figure 2. Cloud services details

The benefits of cloud include: (1) Server/Storage Utilization (2) Self-service (3) Faster Test Provisioning (4) Faster Change Management (5) Speedy Release Management (6) Granular Metering/Billing (7) Fast payback and implementation period for new services (8) Reduced CAPEX (Capital Expenditure.)

In this paper, we are using cloud to provide additional security to the oPass servers. Additionally, it will also help in preventing unauthorized access to the oPass servers as through our mechanism the oPass server becomes totally inaccessible to the user through his/her mobile phone. Currently, it is a convenient method to send Trojans to the user. (Eddy, 2014)(Donohue, 2014).

In oPass, the user sends a HMAC-SHA1 digest (128 bits) of the OTP and other essential fata to the oPass server through a SMS. (Aditya, 2014) We are assuming that the message may be intercepted on its way.

Serpent Key Algorithm (Biham Eli) is a symmetric key block code used for cryptography purposes. It was a finalist in the AES (Advanced Encryption Standard) contest, where it finished in the second place. Designed by Ross Anderson, Eli Biham, and Lars Knudsen, the encryption method has a block size of 128 bits and supports a key size of 128, 192 or 256 bits (Anderson, 2006). The code is a 32-round permutation over a block of 4 32-bit words. Each round applies one of the eight 4-bit to 4-bit S-boxes 32 times in parallel. It was designed in order to operate all operations in parallel, using 32 1-bit slices. Thus, this method not only maximizes parallelism but also allows extensive cryptanalysis work which was performed on DES..

The Serpent Algorithm makers went for a more conservative approach compared to other algorithm designers, electing for a larger security margin compared to the other algorithms. The designers of the algorithm believed 16 rounds to be sufficient against all the known types of attack at that time but specified 32 rounds against future discoveries in cryptanalysis. It should also be noted that the makers of Serpent Algorithm didn't patent it and let it remain in public domain allowing anyone to use it without paying for licensing fees. (Wikipedia, 2002) (Kryptotel.net, 2010) (AES Report, 2012)

2. LITERATURE SURVEY

Our paper is based on the paper, oPass:A User Authentication Protocol Resistant to Password Stealing and Password Reuse Attacks (Sun, 2012) by Hung-Min Sun, Yao-Hsin Chen and Yue-Hsun Lin (2012). As explained earlier, this paper told about the risks and disadvantages of using text passwords. It also told us about the risks in typing passwords into untrusted computers. After this, they propsosed a user authentication protocol called oPass which used a user's cellphone service and SMS (short message service) in order to stop password stealing and reuse attacks. oPass only required that each telecommunication service provider have a unique

phone number. Thus, using this scheme the user had to know only the long term password to login into all the websites.

In the paper Cryptanalysis of oPass (Aditya, 2014) by Shivam Aditya, Varun Mittal and K.Marimuthu, The safety of oPass: A User Authentication Protocol Resistant to Password Stealing and Password Reuse Attacks proposed by H.M.Sun et al. in IEEE Transactions on Information Forensics and Security, Vol.7, No.2, April 2012 was analysed. Upon the full analysis of the paper, four types of attacks, attacks on SMS, unauthorised intruder access using the master password, attacks on oPass communication links, and Network attacks on untrusted web browser were identified in various scenarios. Thus, it was proved that the oPass scheme proposed by H.M.Sun et al. was not ready for practical application.

In the paper, Applying Digital Signature with Encryption Algorithm of User Authentication for Data Security(T. Sivasakthi, 2014) in Cloud Computing by T. Sivasakthi and Dr. N Prabakaran, they told about the cloud computing and it's benefits. They also proposed a user authentication protocol in order to secure data using an encryption algorithm in which they combined data with digital signature in cloud computing. This type of infrastructure ensured the security of information in cloud server.

In the research paper ProcurePass: A User Authentication Protocol to Resist Password Stealing and Password Reuse Attack (Kassim, 2013) by Kassim, M.M. and Sujitha, A., the user tells about the most popular form of user authentication type, the text password. They also tell us about the disadvantages of using text passwords for authentication and the domino effect it can cause. They also explain about the several types of attacks which can be made by typing passwords into kiosks. After telling about these things, they proposed a user authentication protocol named Procedure Pass, which uses a user's mobile phone and short message service (SMS) to prevent password reuse and stealing attacks. It also uses a one-time password (OTP) policy, which frees users from the obligation of remembering all passwords for authentication.

The research paper, Protecting password piracy using authentication protocol (K. Subhashini, 2013) by Subhashini, K. and Bhuvaneswari, M.S. also tell us about the disadvantages and attacks on using text passwords. They also proposed an authentication protocol requiring a web server, an untrusted system and a telecommunication provider. In this, the users only had to remember a long time password too login into all websites.

The research paper named Web based security with LOPass user authentication protocol in mobile application (Bhole, 2013), Bhole, A.T., Chaudhari, S. is also similar to the oPass scheme in its essential and incorporated a similar procedure for user authentication.

In the research paper, Improved Password Authentication System against Password attacks for web Applications (Yalamanchili, 2013) by Vaishnavi Yalamanchili and Dr.P.Pandarinath, they tell about password security, the various attacks in it and the possible steps we should take to prevent it. They also tell about the current trend of banks and companies going for the One time Password (OTP), where users have multiple passwords and use a password only once (in the existing approaches like oPass.) It also tell about the oPass scheme and it's various limitations, for example, on telecommunication service provider and the user's mobile number. In order to improve upon this, they proposed a new method that uses a stronger cryptographic hash function compared to oPass in order to compute secure passwords for multiple accounts while requiring the user to remember only one long term password. Their protocol functioned entirely on the client itself needing no server-side changes. They also implemented an email service in order to recover the user's password after registration.

In the paper, Web Based Security Analysis of oPass Authentication Scheme Using Mobile Application (P. Vaisagan, 2013) by P. Vaisagan, M., Nasreen Fathima and P.Elenthendral, they have told about the entire oPass scheme and have also done the web analysis of the oPass scheme. Various results were found and given in the paper.

The research paper, Enhancing Performance of User Authentication Protocol with Resist to Password Reuse Attacks (R.R.Karthiga, 2012) by Ms. R.R.Karthiga and Mr.K.Aravindhan also tells us about the oPass scheme and its possible advantages to users.

3. PROPOSED METHODOLOGY

In this paper, we have tried to remove the attacks on oPass scheme by impersonating the user. The method used here in order to authenticate the user is using Cryptography along with a 3rd Party Cloud Service Provider.

3.1 The Crypto-Cloud Model

It is an amalgamation of Cryptography and Cloud Computing. It uses Information encryption which is one of the most effective means to achieve cloud computing information security. Its capabilities include providing dual layer of security by combining the features of cryptography along with cloud computing. The

computing model also protects unauthorized access to the oPass servers thus eliminating the risk of any individual trying to directly hack into the oPass server System.

The Crypto-cloud model is based on Symmetric Key Cryptography. In Symmetric Key Cryptography, any symmetric ciphers (Wikipedia, 2002) can be used to encrypt the data along with a key (the digital signature of the user).

In order to encrypt the data along with the signature, we have used the Serpent Algorithm. The serpent cipher algorithm is a variation of block cipher that encrypts a 128 bit block of plaintext by employing a 256 bit key. The algorithm consists of 3 basic functions. These functions are, an initial permutation of bits named IP, a spherical round named R, and a final permutation of bits named FP.

We have used Serpent algorithm instead of the generally used AES (Rijndael). Rijndael won the AES competition primarily because it's the fastest and easiest to implement in hardware, not because it's the most "secure." (AES Report, 2012) Twofish and

Serpent is usually considered more secure, but since they are all quite safe, that is a very individual opinion. And of course, encryption done with multiple algorithms will turn out to be even more "secure," but will reduce the speed even further.

However, as the computer used to check the speed in AES competition was Pentium (200MHz) (Kryptotel.net, 2010), and the speed of computers has increased a lot in the recent years, it has increased the need for the encryption algorithm to be secure.

A graphical speed/security comparison of the different algorithms which were the participants of the AES competition is given below (V Mnssvkr Gupta, 2012). It clearly shows that Serpent Algorithm was proved to be the most secure algorithm in the AES competition.

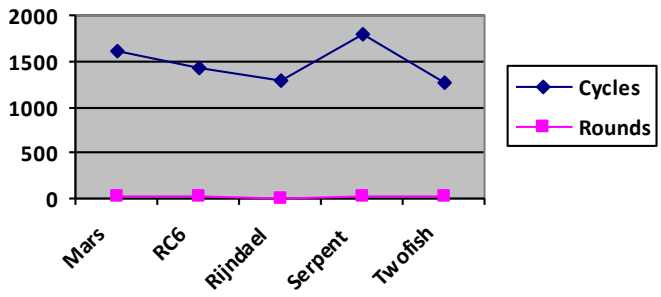

Figure 3. Speed/security comparison of the algorithms of AES

The Encryption Key is the Digital Signature of the system in the algorithm. It can be of 128, 192 or 256 bits, preferably 256bits. We are taking encryption key size as 256 bits.

For the generation of the key, we propose generation of random key using a random number generator (RNG) or pseudorandom number generator (PRNG). (Wikipedia, 2002) A Psuedorandom Number Generator is a type of computer algorithm that generates data which appears to be random when analyzed. Although, the PRNG-generated arrangement is not really random, as it is totally determined by a comparatively small set of starting values, called the PRNG's seed (which may or may not be including fully random values). PRNG's are considered important due to easy reproducibility and fast speed of generation of numbers. (Wikipedia, 2002)

Furthermore, a simple and fast way to execute PRNG (pseudo random number generator) and the applications of it in the field of mobile agent environment has already been proposed earlier. (U. Topaloglu, 2006)

However, this is only the first step.

This key is sent to the oPass server during the registration phase itself in order to use it for decryption by the server.

The oPass server will only be able to be accessed by a trusted 3rd Party Cloud Service Provider. Thus, it totally eliminates the hacking attacks on the oPass servers. The 3rd Party Cloud Service Provider will also only accept the data if sent from the authorized channel (authorized mobile number). This will greatly increase the security on the server side of the service and force the attackers to try and attack from the client side. Thus, we can also easily pinpoint and find the attacker easily if needed.

The current cloud computing structure was mainly developed for data sharing and Security was never the priority of the system. On the contrary, in the crypto-cloud model, security and encryption are inherently integrated. Here, Serpent encrypted data packets are bricks of the whole computing model. Besides having the primary function of doing data encryption and decryption, the crypto cloud model also provides various security

related features. For an example, the exact security leakage can be identified by analyzing the digital signatures of forged data.

3.2 The Serpent Algorithm Encryption Process

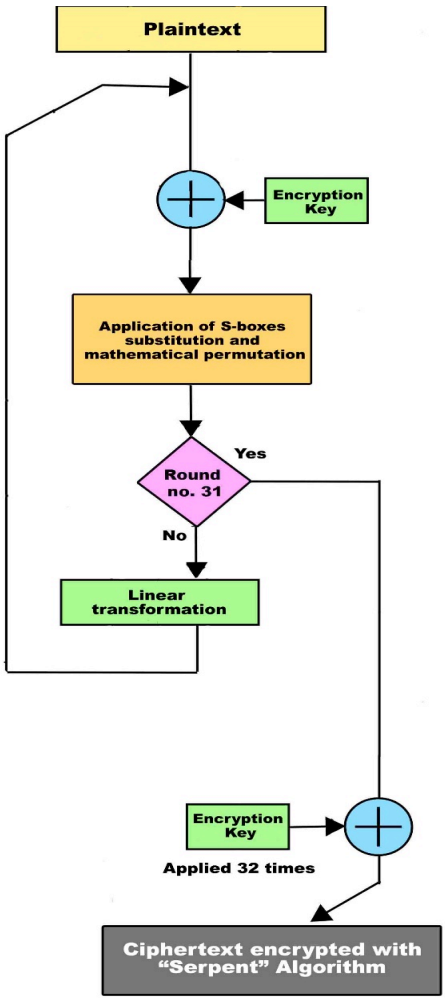

Figure 4. Encryption scheme in Serpent Algorithm

The initial permutation of bits is usually performed by table to decide the position of the bit. However, it may also be performed by replacing the bit at position i with the bit at position (i*32 mod 127), leaving only bits 0 and 127 in place. This permutation's output is labelled B0. The round function is performed 32 times on Bi (starting with B0). In each round (of the function), Bi is first mixed with one of 32 keys (see Key Schedule below for generation notes) using the exclusive-OR operation and then passed through one of eight SBoxes 32 times in parallel. In all but the last round, it is then also subjected to a linear transformation (see the Linear Transformation figure below) to produce Bi+1. In the last round, instead of performing the linear transformation the text is instead mixed with an additional 33rd key.

Once the round functions finishes, a final permutation of bits done in order to place the bits back to the correct position. Like the initial permutation this is frequently done via table lookup but can also be done algorithmically. To perform this operation algorithmically, we need to replace the bit located at position i with the bit located at position (i*4 mod 127), leaving only bits 0 and 127 in place. The result of this final permutation will be the final cipher text of the serpent algorithm.

3.2.1 Key Schedule

In order to perform the Serpent algorithm, we need to generate the round keys from the key provided by the user. The Key Schedule of this algorithm provides 33 128-bit keys to be mixed with the text blocks during the Round function of the algorithm. We need to first create eight 32-bit pre-keys, w-8 to w-1 using the

key provided by the user. The key provided by the user is simply split every 32 bits in order to perform the process. We then produce 132 intermediate keys utilizing the following recurrence:

for i from 0 to 131

wi = (wi-8 xor wi-5 xor wi-3 xor wi-1 xor phi xor i) <<< 11

where phi is the fractional part of the golden ratio (hexadecimal 0x9e3779b9) , xor is the exclusive-or operation and <<< is a left rotation.

We then produce the 33 round keys from the intermediate keys by making them go through the S-Boxes and then combining them into 128-bit blocks, as shown in the following figure:

{k0, k1, k2, k3} := S3 (w0, w1, w2, w3)
{k4, k5, k6, k7} := S3 (w4, w5, w6, w7)
{k8, k9, k10, k11} := S3 (w8, w9, w10, w11)
…
…
{k124, k125, k126, k127} := S3 (w124, w125, w126, w127)
{k128, k129, k130, k131} := S3 (w128, w129, w130, w131)

Sample Input/Output of Serpent Algorithm
KEY: All 0's
PLAIN TEXT: All 0's
CIPHER TEXT: 49 67 2b a8 98 d9 8d f9 50 19 18 4 45 49 10 89

KEY: All 1's
PLAIN TEXT: All 1's
CIPHER TEXT: a4 82 ea a5 d5 77 1f 2f db 2e a1 a5 f1 41 b9 e2

KEY: 00112233445566778899aabbccddeeffffeeddccbbaa99887766554433221100
PLAIN TEXT: 0123456789abcdeffedcba9876543210
CIPHER TEXT: 93 df 9a 3c af e3 87 bd 99 9e eb e3 93 a1 7f ca

3.3 The Serpent Algorithm Decryption Process

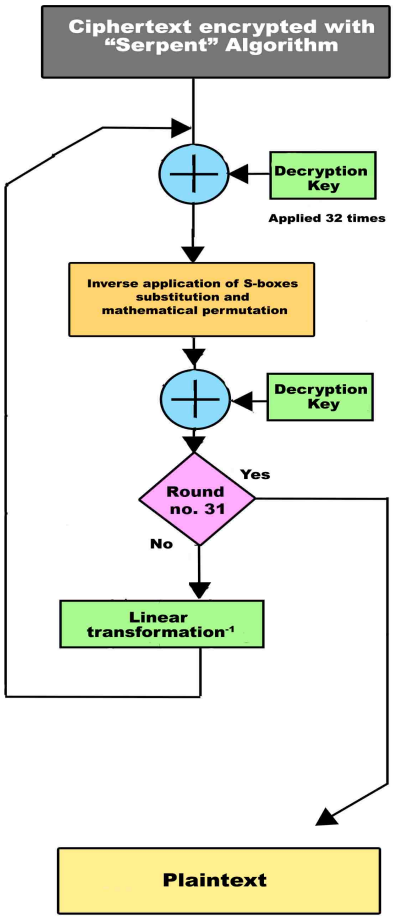

Figure 5. Decryption scheme in Serpent Algorithm

3.4 The Cloud Service Model

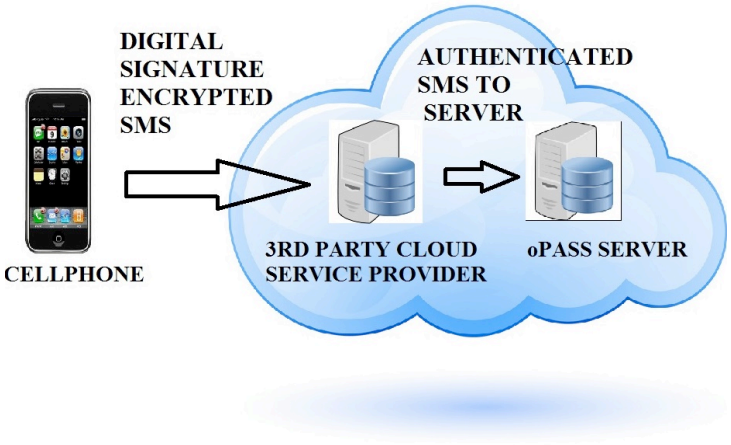

Figure 6. Process flow in the Model

3.5 Step By Step Instructions for the Method

1) The Digital Signature of the system (256 bit) is made using pseudorandom number generator (PNRG) and stored in the local cache. This is the encryption key of the process.

2) This key is also securely sent to the oPass server during the registration phase itself. This is done so that the oPass server can verify the user later on as only the user will have the key to encrypt or decrypt the data.

3) While sending the sms, the 128 bit data (containing OTP and other essential information) is encrypted along with the encryption key (256 bit) using the Serpent Algorithm.

4) The encrypted data is sent using SMS to the trusted 3rd Party Cloud Service Provider through the authorised channel (authorised cell phone number).

5) If authorised, the cloud provider forwards the data to the oPass server.

6) The oPass server then decrypts the data using the decryption key.

7) If the data is found to be useful and authentic, the process is further executed otherwise the authentication mechanism is aborted and the Intrusion warning is sent to the user.

3.6 STRUCTURE OF DATABASE IN CLOUD SERVER

Table 1. Structure of Cloud Database

User ID Number	*User Name*	*Authenticated Mobile Number*
12345	Shivam	9489808518
12346	Dinesh	9656563444
12347	Rahul	9455346365
12348	Keya	9363643635
12349	Ratna	9457474774

4. CONCLUSION

In our earlier paper, we had found many attacks when we had done the cryptanalysis of oPass. In this paper, we have tried to improve upon the existing structure of oPass by eliminating one of these attacks. We have tried to increase the security of the oPass scheme by introducing the concept of digital signatures. The digital signature of the user is encrypted with the data to be sent using the serpent algorithm. The encrypted data is them sent to the oPass servers only through the trusted 3rd party cloud service providers which only forwards the data if it is found clean and sent through authorized channels only. Thus, we have tried to prevent any chance of impersonation of user. Furthermore, it will also protect the oPass server from any malicious attacks as only the trusted 3rd party cloud service provider can contact the server. In our further research papers, we will also try to resolve any possible further attacks which could be made on oPass scheme.

5. REFERENCES

Aditya Shivam, Mittal Varun, K. Marimuthu, D. Ganesh Gopal (2014), Cryptanalysis of oPass, *2014 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT),* 1945-1950

Anderson Ross J. (2006). "Serpent: A Candidate Block Cipher for the Advanced Encryption Standard" (http://www.cl.cam.ac.uk/~rja14 /serpent.html). *University of Cambridge Computer Laboratory*.

Bhole A.T. , Chaudhari S. (2013), Web based security with LOPass user authentication protocol in mobile application, 2013 *IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 1 – 6, ISBN 978-1-4799-1594-1

K. Subhashini, M.S. Bhuvaneswari (2013), Protecting password piracy using authentication protocol*, 2013 International Conference on Information Communication and Embedded Systems (ICICES)*, 116 – 119, ISBN 978-1-4673-5786-9

Kassim M.M, Sujitha A., (2013) ProcurePass: A User Authentication Protocol to Resist Password Stealing and Password Reuse Attack, (ISCBI), *2013 International Symposium on Computational and Business Intelligence*, 31 – 34, ISBN 978-0-7695-5066-4

P. Vaisagan, M., Nasreen Fathima and P.Elenthendral (2013) ,Web Based Security Analysis of oPass Authentication Scheme Using Mobile Application, *IJREAT International Journal of Research in Engineering & Advanced Technology*, Volume 1, Issue 1, March, 2013

R.R.Karthiga, K.Aravindhan (2012), Enhancing Performance of User Authentication Protocol with Resist to Password Reuse Attacks, *International Journal of Computational Engineering Research (ijceronline.com)* Vol. 2 Issue. 8.

Sun Hung-Min, Chen Yao-Hsin, Lin Yue-Hsun (2012), oPass:A User Authentication Protocol Resistant to Password Stealing and Password Reuse Attacks*, IEEE Transactions on Information Forensics and Security 7(2)* 651 – 663

T. Sivasakthi, Dr. N Prabakaran (2014), Applying Digital Signature with Encryption Algorithm of User Authentication for Data Security in Cloud Computing, *International Journal of Innovative Research in Computer and Communication Engineering,* Vol. 2, Issue 2, February 2014

U. Topaloglu, C. Bayrak, K. Iqbal (2006), "A Pseudo Random Number Generator in Mobile Agent Interactions," Engineering of Intelligent Systems, *2006 IEEE International Conference , vol., no., pp.1,5, 0-0 0 doi: 10.1109/ICEIS.2006.1703161*

V Mnssvkr Gupta, K.V.S. Murthy, A.Yesu Babu, R Shiva Shankar (2012), Recent Performance Evaluation Among Various Aes Algorithms - Mars, Rc6, Rijndael, Serpent, Twofish, *International Journal of Science and Advanced Technology (ISSN 2221-8386)*Volume 2 No.2 February 2012

Yalamanchili Vaishnavi, Dr.P.Pandarinath (2013), Improved Password Authentication System against Password attacks for web Applications, *International Journal of Computer Trends and Technology (IJCTT)* – volume 4 Issue 8–August 2013

AES Report (2012), Retrieved 2014.http://csrc.nist.gov/archive/aes/round2/r2report.pdf

Ali Saqib (September 6,2009*) IBM Developer Works*, https://www.ibm.com/developerworks/community/blogs/CloudComputing/entry/nist_s_definition_of_cloud_computing_what_is_cloud_computing?

Biham Eli (n.d.), Retrieved 2014.http://www.cl.cam.ac.uk/~rja14/serpent.html.

Donohue Brian (2014), Retrieved 2014. http://blog.kaspersky.com/fakeinst-targets-us-users/

Eddy Max (2014), Retrieved 2014. http://securitywatch.pcmag.com/mobile-security/323941-mobile-threat-monday-android-trojan-turns-your-phone-into-a-nasty-spam-factory

Kryptotel.net (2010), Retrieved 2014. http://en.kryptotel.net/serpent.html

Wikipedia (2002), Retrieved 2014. http://en.wikipedia.org/wiki/Serpent_(cipher)

Wikipedia (2002), Retrieved 2014. http://en.wikipedia.org/wiki/Symmetric-key_algorithm

Wikipedia (2002), Retrieved 2014. http://en.wikipedia.org/wiki/Key_generation

Wikipedia (2002), Retrieved 2014. http://en.wikipedia.org/wiki/Pseudorandom_number_generator