

Multi-Layered Crypto Cloud Integration of oPass

Shivam Aditya

School of Computer Science and Engineering
VIT University
Vellore, India
sa.shivam.aditya@gmail.com

Varun Mittal

School of Computer Science and Engineering
VIT University
Vellore, India
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 multi-layer crypto-cloud integration with oPass which can handle this kind of attack.

Keywords—Cloud, Impersonation, oPass, RSA, Digital Signature, SMS, Network Security

I. INTRODUCTION

oPass scheme [1] 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

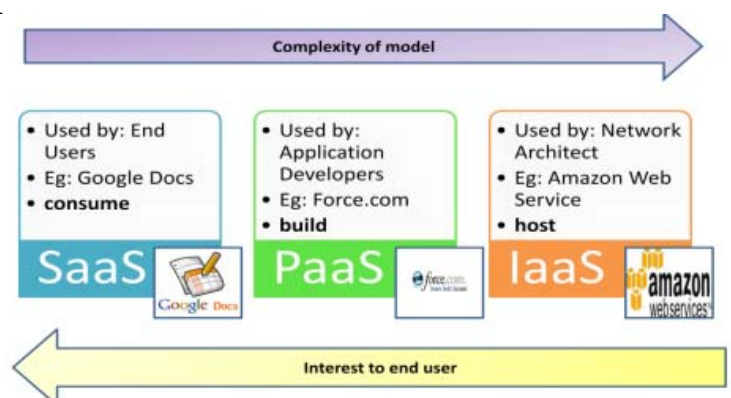
resources (e.g., servers, networks, applications, storage and services) that can be rapidly purveyed and released with minimal effort or service provider interaction.

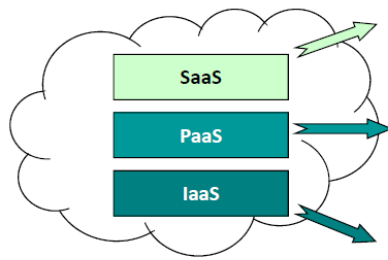
Any user having access to the server can utilize its processing power, store data, or use it in various other purposes. This has given the free will to the user to access the application from anywhere in the world instead of using a personal computer every time to execute a native application.

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. [21]

Its 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.





Who Uses It	What Services are available	Why use it?
Business Users	EMail, Office Automation, CRM, Website Testing, Wiki, Blog, Virtual Desktop ...	To complete business tasks
Developers and Deployers	Service and application test, development, integration and deployment	Create or deploy applications and services for users
System Managers	Virtual machines, operating systems, message queues, networks, storage, CPU, memory, backup services	Create platforms for service and application test, development, integration and deployment

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. [2][3].

In oPass, the user sends a HMAC-SHA1 digest (128 bits) of the OTP and other essential data to the oPass server through a SMS. [7] We are assuming that the message may be intercepted on its way.

Serpent Key Algorithm [17] 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 [4]. 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. [5][16][20]

In our system, we have also used the RSA Algorithm for authentication of both the user and server.

In 1978, Ron Rivest, Adi Shamir, and Leonard Adleman presented a cryptographic algorithm, which was used to replace the less secure National Bureau of Standards (NBS) algorithm. In RSA, a combination of public key cryptosystem and digital signatures is implemented. RSA uses a variable size encryption block and a variable size key [22]. The key-pair is derived from a very large number i.e. n , which is the product of two prime numbers chosen according to special rules; these primes may have length of 100 or more digits each, forming an n with roughly twice as many digits as the prime factors.

RSA algorithm [22] [23]:

1. Select two different prime numbers p and q

For security purposes, the integers p and q must be large.

2. Calculate

$$n = p * q \quad (1)$$

Here, n is used as the module for public key and private key.

3. Calculate

$$f(n) = (q-1)(p-1) \quad (2)$$

where f is a function of Euler's

4. Select an integer e such that $1 < e < f(n)$ and e and $f(n)$ are co-prime. After this, Calculate

$$\text{GCD}(e, f(n)) = 1 \quad (3)$$

5. Determine d : d is multiplicative inverse of

$$e \bmod (f(n)) \quad (e * d) \bmod f(n) = 1 \quad (4)$$

Where d is the private key.

Encryption:

M is plain text data.

$$C = M^e \bmod n \quad (5)$$

Decryption:

C is received cipher text.

$$M = C^d \bmod n \quad (6)$$

II. LITERATURE SURVEY

Our paper is based on the paper, oPass: A User Authentication Protocol Resistant to Password Stealing and Password Reuse Attacks[6] 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 proposed 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[7] by K.Marimuthu, Shivam Aditya and Varun Mittal, 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 analyzed. Upon the full analysis of the

paper, four types of attacks, attacks on SMS, unauthorized 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[8] in Cloud Computing by T. Sivasakthi and Dr. N Prabakaran, they told about the cloud computing and its 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 [9] 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 [10] 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 to login into all websites.

The research paper named Web based security with LOPass user authentication protocol in mobile application [11], 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 [12] 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 tells about the oPass scheme and its 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 [13] 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 [14] by Ms. R.R.Karthiga and Mr.K.Aravindhan also tells us about the oPass scheme and its possible advantages to users.

Veri-Sign [23], an IT company has presented a method to incorporate cloud based OTP service in their security solutions by using RSA Secure ID in their VIP (VeriSign Identity Protection (VIP) Authentication Service).

III. 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.

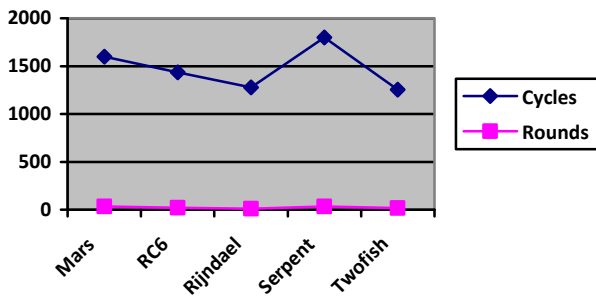
A. The Multi-layered Crypto-Cloud Model

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 [15] 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.

A graphical speed/security comparison of the different algorithms which were the participants of the AES competition is given below [25]. It clearly shows that Serpent Algorithm was proved to be the most secure algorithm in the AES competition.



I Speed/security comparison of the algorithms in 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). [18] A Pseudo random 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 the fast speed of generation of numbers. [19]

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. [6]

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.

Before sending the data to the cloud service provider, the user faces another authentication in order to prove that it's not a bot. At the login page, a trial is presented to the client. This can be a text string, image, a random number etc. The user needs to enter it correctly to continue the process.

When confirmed that the user isn't a bot, the user encrypts the data using Serpent Algorithm with the key generated earlier.

In order to provide additional security, the encrypted data is further encrypted with the server's public key. This data is then sent to the oPass server through the 3rd party cloud service provider.

It is done in order to make the oPass server only accessible by the trusted 3rd Party Cloud Service Provider and remove any external interaction with it. 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.

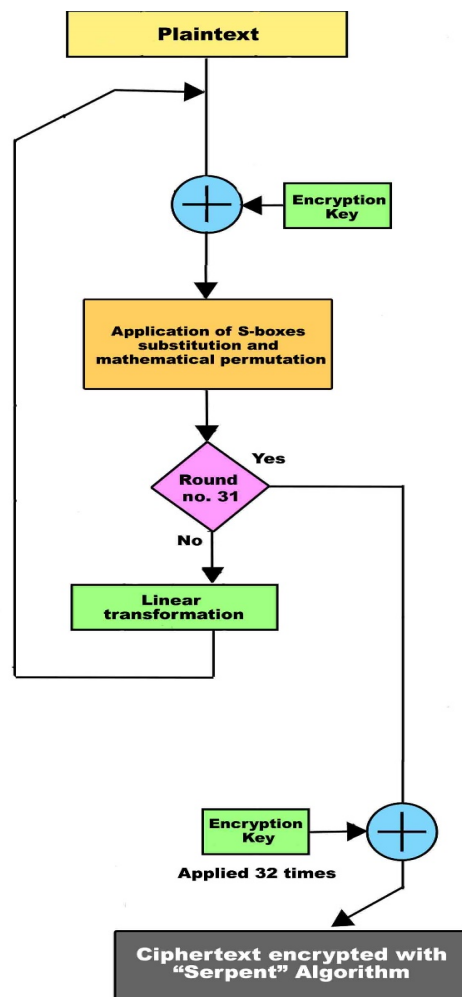
Thus, we can also easily pinpoint and find the attacker easily if needed.

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.

The server then gets the encrypted data from the 3rd party cloud service provider and the decrypts it twice. Once it is done using its own private key for RSA. The next time it is done using the key for the Serpent algorithm which was received earlier. Thus, the server gets the final decrypted data containing the OTP and other essential information. If the OTP is proved to be the same as generated at oPass server, the acknowledgement is sent.

Another change we are proposing is that the acknowledgement of the authentication is sent to the user through both sms and email. The email will be the one the user gives during the registration phase. This email allows for dual notification of successful authentication to the user. This dual notification procedure will have another major benefit for user. In cases where the hacker can hack all the communication devices and keys for the algorithm and is able to successfully impersonate the user, the email will inform the user that someone has successfully been authenticated. Thus, the user will know that there has been a security breach in his system.

B. Serpent Algorithm Encryption Process:



II 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 $(i*32 \bmod 127)$, leaving only bits 0 and 127 in place. This permutation's output is labelled B_0 . The round function is performed 32 times on B_i (starting with B_0). In each round (of the function), B_i 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 B_{i+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 $(i*4 \bmod 127)$, leaving only bits 0 and 127 in place. The result of this final permutation will be the final cipher text of the serpent algorithm.

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
 $w_i = (w_{i-8} \text{ xor } w_{i-5} \text{ xor } w_{i-3} \text{ xor } w_{i-1} \text{ xor } \phi \text{ xor } i) \lll 11$
 where ϕ is the fractional part of the golden ratio (hexadecimal 0x9e3779b9), xor is the exclusive-or operation and \lll is a left rotation.

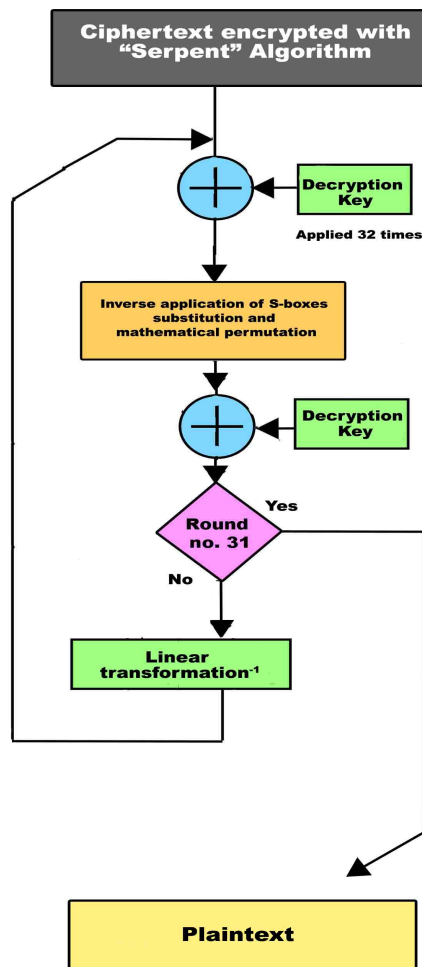
We then produce the 33 round keys from the intermediate keys by making them go through the S-Boxes and then combining them into 128-bit blocks, as shown in the following figure:

$$\begin{aligned}
 \{k_0, k_1, k_2, k_3\} &:= S_3(w_0, w_1, w_2, w_3) \\
 \{k_4, k_5, k_6, k_7\} &:= S_3(w_4, w_5, w_6, w_7) \\
 \{k_8, k_9, k_{10}, k_{11}\} &:= S_3(w_8, w_9, w_{10}, w_{11}) \\
 &\dots \\
 \{k_{124}, k_{125}, k_{126}, k_{127}\} &:= S_3(w_{124}, w_{125}, w_{126}, w_{127}) \\
 \{k_{128}, k_{129}, k_{130}, k_{131}\} &:= S_3(w_{128}, w_{129}, w_{130}, w_{131})
 \end{aligned}$$

Sample Input/Output of Serpent Algorithm

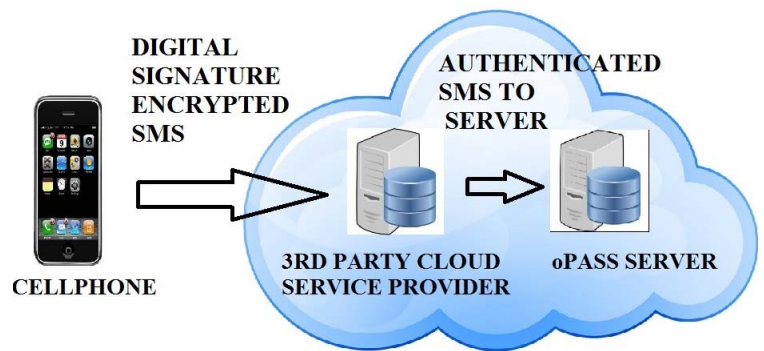
- KEY: All 0's
 PLAIN TEXT: All 0's
 CIPHER TEXT: 49 67 2b a8 98 d9 8d f9 50 19 18 4
 45 49 10 89
- KEY: All 1's
 PLAIN TEXT: All 1's
 CIPHER TEXT: a4 82 ea a5 d5 77 1f 2f db 2e a1 a5
 f1 41 b9 e2
- KEY:
 00112233445566778899aabbccddeeffffeedddccbbaa9
 9887766554433221100
 PLAIN TEXT:
 0123456789abcdeffedcba9876543210
 CIPHER TEXT: 93 df 9a 3c af e3 87 bd 99 9e eb e3
 93 a1 7f ca

C. Serpent Algorithm Decryption Process:



III Decryption scheme in Serpent Algorithm

D. The Cloud Service Model:



IV PROCESS FLOW

E. 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) Before sending the data, the user faces another authentication in order to prove that it's not a bot. At the login page, a trial is presented to the client. This can be a text string, image, a random number etc. The user needs to enter it correctly to continue the process.

4) 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.

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

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

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

8) 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.

F. STRUCTURE OF DATABASE IN CLOUD SERVER

TABLE I.

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

Fig. 1. Structure of Cloud Database

IV. 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 then 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.

REFERENCES

- [1] Hung-Min Sun, Yao-Hsin Chen, and Yue-Hsun Lin (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
- [2] <http://securitywatch.pcmag.com/mobile-security/323941-mobile-threat-monday-android-trojan-turns-your-phone-into-a-nasty-spam-factory>
- [3] <http://blog.kaspersky.com/fakeinst-targets-us-users/>
- [4] Ross J. Anderson (2006-10-23). "Serpent: A Candidate Block Cipher for the Advanced Encryption Standard" (<http://www.cl.cam.ac.uk/~rja14/serpent.html>). University of Cambridge Computer Laboratory. Retrieved 2013-01-14.
- [5] [http://en.wikipedia.org/wiki/Serpent_\(cipher\)](http://en.wikipedia.org/wiki/Serpent_(cipher))
- [6] Topaloglu, U.; Bayrak, C.; Iqbal, K., "A Pseudo Random Number Generator in Mobile Agent Interactions," Engineering of Intelligent Systems, 2006 IEEE International Conference on , vol., no., pp.1,5, 0-0 0 doi: 10.1109/ICEIS.2006.1703161
- [7] Cryptanalysis of oPass by K.Marimuthu, D.Ganesh Gopal, Shivam Aditya, Varun Mittal, 2014 IEEE International Conference on Advanced Communication Control and Computing Technologies (ICACCCT) 1945-1950
- [8] Applying Digital Signature with Encryption Algorithm of User Authentication for Data Security in Cloud Computing, T. Sivasakthi and Dr. N. Prabhakaran, International Journal of Innovative Research in Computer and Communication Engineering, Vol. 2, Issue 2, February 2014
- [9] ProcurePass: A User Authentication Protocol to Resist Password Stealing and Password Reuse Attack, Kassim, M.M. and Sujitha, A., (ISCBI), 2013 International Symposium on Computational and Business Intelligence, 31 – 34, ISBN 978-0-7695-5066-4
- [10] Protecting password piracy using authentication protocol, Subhashini, K., Bhuvaneswari, M.S., 2013 International Conference on Information Communication and Embedded Systems (ICICES), 116 – 119, ISBN 978-1-4673-5786-9
- [11] Web based security with LOPass user authentication protocol in mobile application, Bhole, A.T. , Chaudhari, S., 2013 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 1 – 6, ISBN 978-1-4799-1594-1
- [12] Improved Password Authentication System against Password attacks for web Applications, Vaishnavi Yalamanchili, Dr.P.Pandarinath, International Journal of Computer Trends and Technology (IJCTT) – volume 4 Issue 8–August 2013
- [13] Web Based Security Analysis of oPass Authentication Scheme Using Mobile Application by P. Vaisagan, M., Nasreen Fathima and P.Elenthendral, IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 1, March, 2013
- [14] Enhancing Performance of User Authentication Protocol with Resist to Password Reuse Attacks by Ms. R.R.Karthiga and Mr.K.Aravindhan, International Journal of Computational Engineering Research (ijceronline.com) Vol. 2 Issue. 8.
- [15] http://en.wikipedia.org/wiki/Symmetric-key_algorithm
- [16] <http://en.kryptotel.net/serpent.html>
- [17] <http://www.cl.cam.ac.uk/~rja14/serpent.html>
- [18] http://en.wikipedia.org/wiki/Key_generation
- [19] http://en.wikipedia.org/wiki/Pseudorandom_number_generator
- [20] <http://csrc.nist.gov/archive/aes/round2/r2report.pdf>
- [21] IBM, https://www.ibm.com/developerworks/community/blogs/CloudComputing/entry/nist_s_definition_of_cloud_computing_what_is_cloud_computing?
- [22] Atul Kahate: Cryptography and Network Security, Tata McGraw –Hill Publishing Company Limited (2003)
- [23] RSA and VeriSign, <http://webcache.googleusercontent.com/search?q=cache:http://www.channelinsider.com/c/a/Cloud-Computing/RSA-and-VeriSign-Partner-on-CloudBased-OTP-Service-146773>
- [24] William Stallings, "Cryptography and Network Security Principles and Practices", Prentice Hall, New Delhi. (2005)
- [25] RECENT PERFORMANCE EVALUATION AMONG VARIOUS AES ALGORITHMS - MARS, RC6, RIJNDAEL, SERPENT, TWOFISH by V. MNSSVKR GUPTA, K.V.S. MURTHY, DR.A. YESU BABU, R. SHIVA SHANKAR International Journal of Science and Advanced Technology (ISSN 2221-8386) Volume 2 No.2 February 2012