A Novel Password Protected Key Exchange Protocol

Ravi Kiran Labhala[#1], Santosh Naidu P[#2]

[#1,2]*Computer Science and Engineering, MVGR College of Engineering, India*

[1]ravi77.2011@gmail.com

[2]amsan2015@gmail.com

***Abstract*— Exchanging messages are more common thing lately. More number of people connects with each other in the network and (verifies someone's identity) each other while sharing their data. So users following so many rules of conduct for providing security to their data and the servers which they are storing their data. Due to all data storing in the single server, there is a chance to hack server data to be told (to people). This paper presents a solution to this problem such as (verifying someone's identity) process has to share by two servers. Client has to (verify someone's identity) in two servers like two step checking (for truth). It also includes (related to secret computer codes) ways of doing things to provide security for the data stored in the servers.**

***Keywords*— diffie-hellman key exchange protocol, elgamal digital signature, AES, Rijndael's key schedule, Bellovin and Merritt**

I. General Terms

Security, Cryptography

II. Introduction

Bellovin and Merritt were the first to think about (verifying someone's identity) based on password only, and introduced a set of so-called "(turned into secret code) key exchange" rules of conduct, where the password is used as a secret key to (turn into secret code) random numbers for key exchange purpose. Formal models of security for the password-only (verifying someone's identity) were first given independently by Bellare et al. and Boyko et al. Katz et al. were the first to give a password-only (verifying someone's identity) rules of conduct which is both practical and provably secure under standard (related to secret computer codes) idea (you think is true). Next method the researchers introduced a rules of conduct that includes diffie-hellman key exchange rules of conduct, and elgamal digital signature in the (verifying someone's identity) process which is called as password (verifying someone's identity) key exchange rules of conduct.

In the study of the existing methods (related to secret computer codes) ways of doing things and key exchange rules of conduct are very basic rules of conduct which are easily breakable in the networks and less security. In the elgamal big plan/layout/dishonest plan there is problem with the hash sets of computer instructions leads to more calculation complex difficulty and the takes more processing time. So we introduced a highly secured. In used (having a left half that's a perfect mirror image of the right half) password based key exchanging rules of conduct and Advanced Standard (turning messages into secret code) (related to secret computer codes) big plan/layout/dishonest plan and random value based method is used to provide security for the data and the client (verifying someone's identity). In these two servers maintains this (verifying someone's identity) and (the science of making secret codes). One server for initial (verifying someone's identity) and another server for next (verifying someone's identity) and (the science of making secret codes). By using we can reduce more hacking problems in the network. Client and servers share the public property and communicate with each other.

III. Literature Survey

In 2003, John Brainard, Ari Juels, Burt Kaliski, and Michael Szydlo described a new, two-server secure roaming system that benefits from an especially lightweight new set of rules of conduct. In contrast to previous ideas, ours can be put into use to require (almost completely) no intensive (related to secret computer codes) computation by clients. This and other design features make/give the system, in our view, the most practical proposal to date in this area. We describe in this paper the rules of conduct and putting into use challenges and the design choices hidden (under) the system.[1]

In 2003, Mario Di Raimondo and Rosario Gennaro proposed first rules of conduct which are provably secure in the standard model (i.e. no random magicians are used for the proof of security). More than that our rules of conduct are reasonably efficient and implementable in practice. In particular a goal of the design was to avoid expensive zero-knowledge proofs to keep interaction to a very low value.[2]

In 2000, Warwick Ford and Burton S. Kaliski Jr. said that a roaming user, who accesses a network from different client terminals, can be supported by a (written proof of identity, education, etc.) server that (verifies someone's identity) the user by password then helps in launching a secure environment for the user. However, traditional (written proof of identity, education, etc.) server designs are able to be hurt by thorough password guessing attack at the server. We describe a new (written proof of identity, education, etc.) server model and supporting rules of conduct that overcomes that (not having enough of something). The rules of conduct provides for securely creating a strong secret from a weak secret (password), based on communications exchanges with two or more independent servers. The result can be (including a lot of debt) in different ways, for example, the strong secret can be used to (change secret codes into readable messages) an secret/unreadable private key or it can be used in strongly (verifying someone's identity) to an application server. The rules of conduct has the properties that a would-be attacker cannot feasibly figure out/calculate the strong secret and has only a limited opportunity to guess the password, even if he or she has access to all messages and has control over some, but not all, of the servers.[3]

Jonathan Katz, Rafail Ostrovsky and Moti Yung3 showed an efficient, 3-round, password-authenticated key exchange protocol with human-memorable passwords which is provably secure under the Decisional Diffie-Hellman idea (you think is true), yet needs/demands only (roughly) 8 times more computation than \standard" Diffiee-Hellman key exchange (which provides no (verifying someone's identity) at all). We assume public parameters available to all parties. We stress that we work in the standard model only, and do not require a \random magician" idea (you think is true).[4]

Jonathan Katz, Philip MacKenziey,Gelareh Tabanz and Virgil Gligorx showed that a two-server version of the password-only key-exchange rules of conduct of Katz, Ostrovsky, and Yung (the KOY rules of conduct). Our work gives the first secure two-server rules of conduct for the password-only setting (in which the user need remember only a password, and not the servers' public keys), and is the first two-server rules of conduct (in any setting) with a proof of security in the standard model. Our work this way fills a gap left by the work of MacKenzie et al. (J. Crypto 2006) and Di Raimondo and Gennaro (JCSS 2006). As an added/more benefit of our work, we show modifications that improve the efficiency of the original KOY rules of conduct.[5]

In 2005 Philip MacKenzie and Thomas Shrimpton proposed an efficient password-authenticated key exchange system involving a set of servers with known public keys, in which a certain threshold of servers must participate in the authentication of a user, and in which the compromise of any fewer than that threshold of servers does not allow an attacker to perform an offline dictionary attack. We prove our system is secure in the random oracle model under the Decision Diffie-Hellman assumption against an attacker that may eavesdrop on, insert, delete, or modify messages between the user and servers, and that compromises fewer than that threshold of servers.[6]

Yanjiang Yang,Feng Bao andRobert H. Deng proposed the rapid rise of federated enterprises entails a new way of trust management by the fact that the enterprise can account for partial trust of its affiliating organizations. On the other hand, password has historically been used as a main means for user authentication because of operational simplicity. We are thus motivated to explore the use of short password for user authentication and key exchange in the context of federated enterprises. Exploiting the special structure of a federated enterprise, our proposed new architecture comprises an external server (managed by each affiliating organization) and a central server (managed by the enterprise headquarter). We are concerned with the development of an efficient authentication and key exchange protocol using password, built over the new architecture. The architecture together with the protocol well addresses online dictionary attacks initiated at the server side, a problem rarely considered in prior effort.[7]

In 2006, Yanjiang Yang, Robert H. Deng, Senior Member, and Feng Bao proposed that only a front-end service server engages directly with users while a control server stays behind the scene; therefore, it can be directly applied to strengthen existing single-server password systems. In addition, the system is secure against offline dictionary attacks mounted by either of the two servers.[8]

In 2006, Yanjiang Yang, Robert H. Deng and Feng Bao generalized the two-server model to an (related to the beautiful design and construction of buildings, etc.) of a single control server supporting multiple service servers, tailored to the organizational structure of IDSs. The hidden (under) user (verifying someone's identity) and key exchange rules of conduct we propose are password-only, neat, efficient, and strong and healthy against off-line dictionary attacks mounted by both servers.[9]

IV. EXISTING SYSTEM

Security Services

For the third party auditing in cloud storage systems, there are several important requirements which have been proposed in some previous works. The auditing protocol should have the following properties:

1) Confidentiality: The auditing protocol should keep owner’s data confidential against the auditor.

2) Dynamic Auditing: The auditing protocol should support the dynamic updates of the data in the cloud.

3) Batch Auditing: The auditing protocol should also be able to support the batch auditing for multiple owners and multiple clouds.

Problem Statement

In the auditing process there may be chance to leak the received data. There may be chance to following attacks: Replay attack, Forge attack and Replace attack.

1) Replace Attack: The server may choose another valid and uncorrupted pair of data block and data tag (mk, tk) to replace the challenged pair of data block and data tag (mi, ti), when it already discarded mi or ti.

2) Forge Attack: The server may forge the data tag of data block and deceive the auditor, if the owner’s secret tag keys are reused for the different versions of data.

3) Replay Attack: The server may generate the proof from the previous proof or other information, without retrieving the actual owner’s data.

The main challenge in the design of data storage auditing protocol is the data privacy problem (i.e., the auditing protocol should protect the data privacy against the auditor.). This is because:

1) For public data, the auditor may obtain the data information by recovering the data blocks from the data proof.

2) For encrypted data, the auditor may obtain content keys somehow through any special channels and could be able to decrypt the data. To solve the data privacy problem, our method is to generate an encrypted proof with the challenge stamp by using the Bi-linearity property of the bilinear pairing,

such that the auditor cannot decrypt it. But the auditor can verify the correctness of the proof without decrypting it.

In existed system many of the algorithms encrypting the plain text to cipher text. But the algorithms applying same encryption process to entire plain text. So if the same type of characters repeated in plain text, that all characters converting into the same type of cipher text. The cryptanalysis for this type of cipher texts is becoming easy process. For example if the plain text is "BANANA". In this plain text „A‟ is repeated 3 times and „N‟ is repeated 2 times. In the present existed algorithms 3A‟s and 2N‟s will be encrypted in to same characters. In decryption 3 characters is enough to get this plain text. For those texts cryptanalysis will become easy for these type plain texts.

Disadvantages

1. By increasing scalability there increase in work load in server to maintain, authenticate, and store the data securely.
2. By using public key cryptographic techniques in encoding the data there is leakage of data in data packets.
3. By low bandwidths and low performance of the server security should me hard task.

V. PROPOSED SYSTEM

In the study of the existing methods cryptographic techniques and key exchange protocol are very basic protocols which is easily breakable in the networks and less security. In the elgamal scheme there is problem with the hash algorithms leads to more calculation complexity and the takes more processing time. So we introduced a highly secured. In used Symmetric password based key exchanging protocol and Advanced Standard Encryption cryptographic scheme and random value based method is used to provide security for the data and the client authentication. In these two servers maintains this authentication and cryptography. One server for initial authentication and another server for next authentication and cryptography. By using we can reduce more hacking problems in the network. Client and servers share the public property and communicate with each other.

Algorithm

For every user in the system have to register in two servers. In this process we designed a method that is random value based protocol that processes the exchange between the users and the first server.

Input: user Id 'UId', random number 'R'

Steps:

Step 1: User select a random number R and sends to server S1.

Step 2: Sever S1 generates key.

a. Sever selects a random numbers Q1 and random numbers Q2

b. Send these two random numbers send to User 'UId'.

Step3: User reveals the secret key Sk=(R*Q1)+Q2

After generating this key user uses this key for authentication and encrypting the text.

Step-1:

It is Symmetric block cipher

Block length: 128 bits (P = C = {0,1}1 2 8)

Key lengths: 128, 192, 256 bits (K = {0,1}1 2 8 , ...)

At least as secure as Triple-DES, but more efficient

Step-2:

KeyExpansions—round keys are derived from the cipher key using Rijndael's key schedule.

AES requires a separate 128-bit round key block for each round plus one more.

Step-3:

InitialRound

Add Round Key—each byte of the state is combined with a block of the round key using bitwise xor.

Step-4:

Rounds

a. SubBytes: a non-linear substitution step where each byte is replaced with another according to a lookup table.

b. ShiftRows: a transposition step where the last three rows of the state are shifted cyclically a certain number of steps.

c. MixColumns:a mixing operation which operates on the columns of the state, combining the four bytes in each column.

Step-5:

Add Round Key

Final Round (no MixColumns)

a. Sub Bytes: The SubBytes() transformation is a non-linear byte substitution that operates independently on each byte of the State

b. Shift Rows:

In the ShiftRows() transformation, the bytes in the last three rows of the State are cyclically shifted over different numbers of bytes (offsets).

The MixColumns() transformation operates on the State column-by-column, treating each column as a four-term polynomial as described in Sec. 4.3. The columns are considered as polynomials over GF(28) and multiplied modulo x 4 + 1 with a fixed polynomial a(x), given by

a(x) = {03}x 3 + {01}x 2 + {01}x + {02} .

Advantages

1. By using two servers in we can reduce the workload of the server.
2. It increases the performance of the server and the security of the data
3. I can increases the efficiency of the storage as well as the retrieval of the data from the server.
4. By using secure code verification it provides double security of authentication.

Requirements

1. Operating system: Microsoft Windows XP, Windows-7,Windows-8
2. CPU: 32 bit or 64 bit processor
3. System memory: minimum of 512 MB RAM
4. Storage: 100 MB of available hard-disk space

5. Run Time - .Net Framework 3.0 or above
6. Eclipse

VI. PROCESS IMPLEMENTATION

Here we provide the process of implementation along with screenshots.

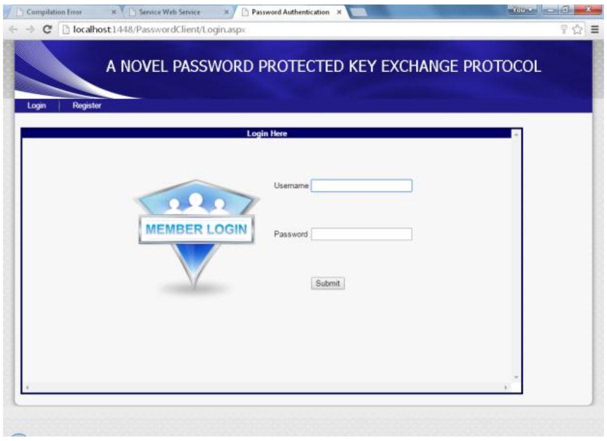
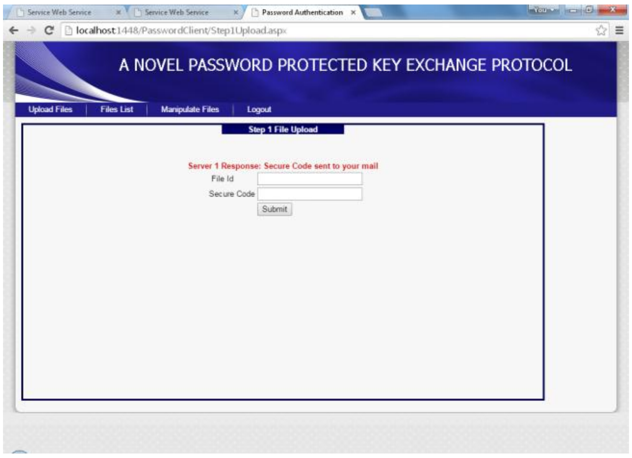


Figure 1 Login Interface



Server 1 Response: Secure Code sent to your mail

Figure 2 Authentication

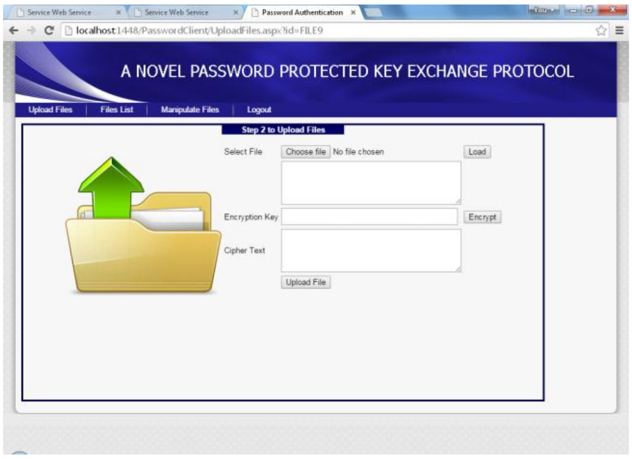
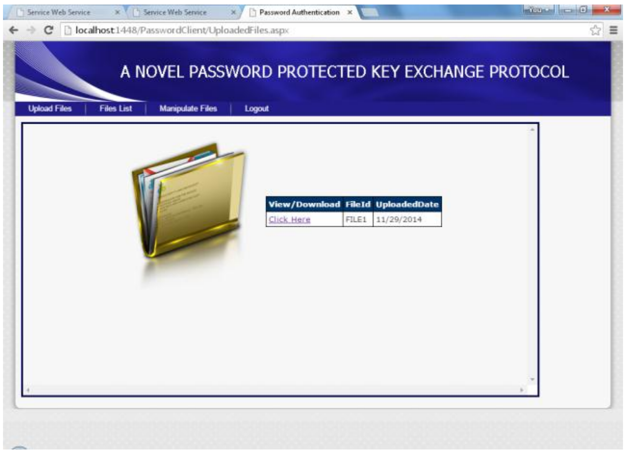
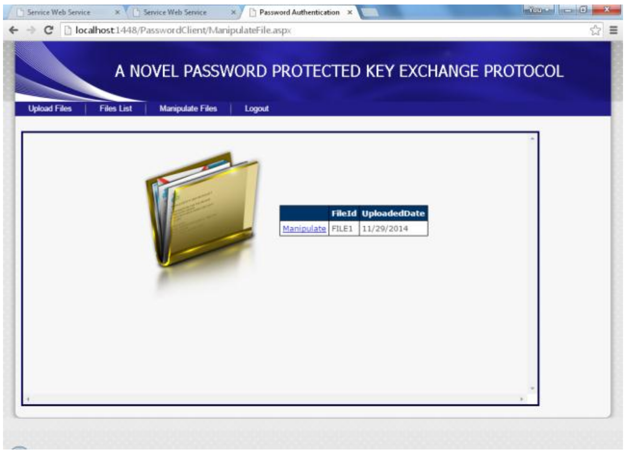


Figure 3 File Upload and Encryption Interface



View/Download	FileId	UploadedDate
Click Here	FILE1	11/29/2014

Figure 4 File List Interface



	FileId	UploadedDate
Manipulate	FILE1	11/29/2014

Figure 5 File Manipulation Interface

VII. TEST CASES

Table 1 Test Cases of the process

Test No.	Test Case	Expected Output	Actual Output	Result
1.	Invalid Log In Test: By providing invalid User name and Password	A dialog Box to be displayed saying Invalid Login, Access Denied	A dialog Box is displayed saying Invalid Login, Access Denied	Passed
2.	Valid Log In Test: By providing Valid User name and Password	The Text Screen for accepting the text to be shown	The Text Screen for accepting the text is shown	Passed

VIII. CONCLUSION AND FUTURE WORK

In this project we proposed framework that combines with cryptographic properties with secure storage. Our framework introduces secure data auditing for multiple owners and Upload their data in third party server. By using our method we can reduce the work load of the authentication and the storage services. By using cryptographic techniques and secure code authentication we increased the guarantee for the security of the data and the database.

IX. ACKNOWLEDGEMENTS

I would like to thank Department of Computer Science and Engineering staff and management of MVGR College of Engineering for their support in doing this work.

REFERNECES

[1] M. Armbrust et al., "Above the Clouds: A Berkeley View of Cloud Computing," technical report, Univ. of California, Feb. 2009.

[2] S. Das, D. Agrawal, and A.E. Abbadi, "Elastras: An Elastic Transactional Data Store in the Cloud," Proc. Conf. Hot Topics in Cloud Computing (USENIX HotCloud '09), 2009.

[3] D.J. Abadi, "Data Management in the Cloud: Limitations and Opportunities," IEEE Data Eng. Bull., vol. 32, no. 1, pp. 3-12, Mar. 2009.

[4] A.J. Lee and M. Winslett, "Safety and Consistency in Policy-Based Authorization Systems," Proc. 13th ACM Conf. Computer and Comm. Security (CCS '06), 2006.

[5] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - Ocsp," RFC 2560, http://tools.ietf.org/html/rfc5280, June 1999.

[6] E. Rissanen, "Extensible Access Control Markup Language (Xacml) Version 3.0," http://docs.oasis-open.org/xacml/3.0/ xacml-3.0-core-spec-os-en.html, Jan. 2013.

[7] D. Cooper et al., "Internet x.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 5280, http://tools.ietf.org/html/rfc5280, May 2008.

[8] J. Li, N. Li, and W.H. Winsborough, "Automated Trust Negotiation Using Cryptographic Credentials," Proc. 12th ACM Conf. Computer and Comm. Security (CCS '05), Nov. 2005.

[9] L. Bauer et al., "Distributed Proving in Access-Control Systems," Proc. IEEE Symp. Security and Privacy, May 2005.

[10] J. Li and N. Li, "OACerts: Oblivious Attribute Based Certificates," IEEE Trans. Dependable and Secure Computing, vol. 3, no. 4, pp. 340- 352, Oct.-Dec. 2006.

[11] J. Camenisch and A. Lysyanskaya, "An Efficient System for Non-Transferable Anonymous Credentials with Optional Anonymity Revocation," Proc. Int'l Conf. Theory and Application of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT '01), 2001.

[12] P.K. Chrysanthis, G. Samaras, and Y.J. Al-Houmaily, "Recovery and Performance of Atomic Commit Processing in Distributed Database Systems," Recovery Mechanisms in Database Systems, Prentice Hall PTR, 1998.

[13] M.K. Iskander, D.W. Wilkinson, A.J. Lee, and P.K. Chrysanthis, "Enforcing Policy and Data Consistency of Cloud Transactions," Proc. IEEE Second Int'l Workshop Security and Privacy in Cloud Computing (ICDCS-SPCCICDCS-SPCC), 2011.

[14] G. DeCandia et al., "Dynamo: Amazons Highly Available Key-Value Store," Proc. 21st ACM SIGOPS Symp. Operating Systems Principles (SOSP '07), 2007.

[15] F. Chang et al., "Bigtable: A Distributed Storage System for Structured Data," Proc. Seventh USENIX Symp. Operating System Design and Implementation (OSDI '06), 2006.

[16] A. Lakshman and P. Malik, "Cassandra- A Decentralized Structured Storage System," ACM SIGOPS Operating Systems Rev., vol. 44, pp. 35-40, Apr. 2010.

[17] B.F. Cooper et al., "PNUTS: Yahoo!'s Hosted Data Serving Platform," Proc. VLDB Endowment, vol. 1, pp. 1277-1288, Aug. 2008.

[18] W. Vogels, "Eventually Consistent," Comm. ACM, vol. 52, pp. 40- 44, Jan. 2009.

[19] H. Guo, P.-A. Larson, R. Ramakrishnan, and J. Goldstein, "Relaxed Currency and Consistency: How to Say "Good Enough" in SQL," Proc. ACM Int'l Conf. Management of Data (SIGMOD '04), 2004.

[20] T. Kraska, M. Hentschel, G. Alonso, and D. Kossmann, "Consistency Rationing in the Cloud: Pay Only When It Matters," Proc. VLDB Endowment, vol. 2, pp. 253-264, Aug. 2009.

[21] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," Proc. 14th ACM Conf. Computer and Comm. Security (CCS '07), 2007.

[22] K.D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," Proc. 16th ACM Conf. Computer and Comm. Security (CCS '09), 2009.

[23] A. Shraer, C. Cachin, A. Cidon, I. Keidar, Y. Michalevsky, and D. Shaket, "Venus: Verification for Untrusted Cloud Storage," Proc. ACM Workshop Cloud Computing Security (CCSW '10), 2010.

[24] P. Williams, R. Sion, and B. Carbunar, "Building Castles Out of Mud: Practical Access Pattern Privacy and Correctness on Untrusted Storage," Proc. 15th ACM Conf. Computer and Comm. Security (CCS '08), 2008.

[25] P. Williams, R. Sion, and D. Shasha, "The Blind Stone Tablet: Outsourcing Durability to Untrusted Parties," Proc. 16th Annual Network and Distributed System Security Symp. (NDSS '09), 2009.

[26] Z. Wei, G. Pierre, and C.-H. Chi, "Scalable Transactions for Web Applications in the Cloud," Proc. 15th Int'l Euro-Par Conf. Parallel Processing (Euro-Par '09), Aug. 2009.

[27] T. Wobber, T.L. Rodeheffer, and D.B. Terry, "Policy-Based Access Control for Weakly Consistent Replication," Proc. ACM Fifth European Conf. Computer Systems (EuroSys '10), 2010.