International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 04 | Apr-2016 www.irjet.net p-ISSN: 2395-0072

Cloud Computing Approach using Threshold Cryptography for Outsourced Data

GargiWalse Patil[1],BhagyshreeShinde[2],Amruta Shinde[3],ApurvaPawar[4],Prof. B.R. Burghate[5]

[1234]UG Student, SPPU, BhivarabaiSawant Institute of Technology & Research, Pune, India

[5]Asst.Professor, SPPU, BhivarabaiSawant Institute of Technology & Research, Pune, India

---

Abstract—*This paper has two public-key schemes to achieve deniable authentication for the Internet Key Exchange (IKE). Authentication in security had is the essential factor in the key establishment over Internet. The Deniable Internet key exchange protocol gives more value to the IKE standard. Our schemes can in some situations be more efficient than existing IKE protocols as well as having stronger deniability properties. Key-exchange, in particular Diffie–Hellman key exchange (DHKE), is among the core cryptographic mechanisms for ensuring network security. For key-exchange over the Internet, both security and privacy are desired. In this paper, we develop a family of privacy-preserving authenticated DHKE protocols named deniable Internet key-exchange (DIKE), both in the traditional PKI setting and in the identity-based setting. The newly developed DIKE protocols are of conceptual clarity and practical (online) efficiency. They provide useful privacy protection to both protocol participants, and add novelty and new value to the IKE standard.*

In this scheme, there are basically three entities: Data Owner (DO), Cloud Service Provider (CSP) and Users.

Key Words— Authentication, Diffie–Hellman, key exchange, security, privacy, deniability

1. INTRODUCTION

Cryptography or cryptology, "hidden, secret", "writing", "study", respectively is the practice and study of techniques for secure communication in the presence of third parties (called adversaries). More generally, it is about constructing and analysing protocols that block adversaries; various aspects in information security such as dataconfidentiality, data integrity, authentication, and non-repudiation are central to modern cryptography. Modern cryptography exists at the intersection of the disciplines of mathematics, computer science, and electrical engineering. Applications of cryptography include ATM cards, computer passwords, and commerce. Cryptography prior to the modern age was effectively synonymous with encryption, the conversion of information from a readable state to apparent nonsense.

In cryptography, a cryptosystem is called a "threshold cryptosystem", if in order to decrypt an encrypted message, several parties (more than some threshold number) must coop-erate in the decryption protocol. The message is encrypted using a public key and the corresponding private key is shared among the participating parties. Let be the number of parties. Such a system is called (t,n)-threshold, if at least of these parties can efficiently decrypt the ciphertext, while less than t have no useful information. Similarly it is possible to define (t,n)-threshold signature scheme, where at least parties are required for creating a signature. Threshold versions of encryption schemes can be built for many public encryption schemes. The natural goal of such schemes is to be as secure as the original scheme. The two basic security requirements for outsourced data in cloud computing is data confidentiality and access control. While seeing the security of data we also have to take concern about performance of system which include DO,CSP,Users.

In group-key scheme, there is single key which corresponds to each group of users for the process of decryption and all users of that particular group are familiar with that key. With this scheme data security and performance is upgraded.

Objectives:

- Data Confidentiality
- Entity Authentication
- Data Integrity
- Data Access Control

II. System Architecture

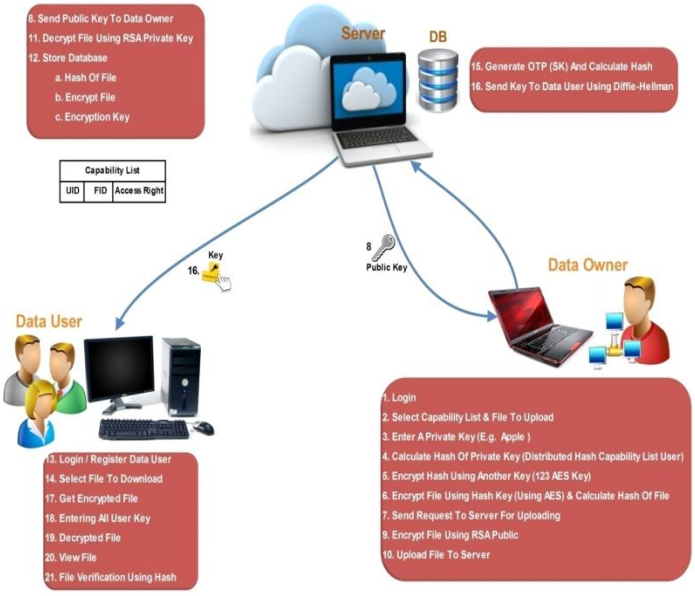


Figure 1:System Architecture

The explanation of architecture model is describes the process what CSP do after getting encrypted data and Capability List from the DO. CSP decrypts the message using its own private key and the public key of data owner and stores the encrypted data and Capability List in its storage. CSP then updates the encrypted File List and Capability List. Since, data are encrypted using symmetric key (KT) which is known only to DO and respected user group, CSP can't see data even though user's credential comes through it. Algorithm 2 illustrates the procedure required after a new File creation. When a new File is created, DO fills entries for that File in Capability List containing UID, FID and AR. DO generates a symmetric key (KT) and encrypts File with that symmetric key (KT). Now, DO encrypts the updated CPList, Encrypted File and symmetric key (KT) with its private key after that public key of CSP and sends these to the CSP. When CSP receives these, it updates Capability List, Encrypted File List and sends encrypted symmetric key (KT) to respective user group. Users of the user group then decrypt the message and get their own parts of the symmetric key (KT). To avoid man-in-middle and replay attack we use nonce and timestamp in each message. After getting the details, user can request to CSP for data. Algorithm 3 describes how data are exchanged securely between CSP and the user by use of modified Diffie Hellman algorithm. We called it modified D-H algorithm as we encrypt the D-H parameters using the public key of one side and, using nonce in each direction during session key (KS) generation and data transfer. It helps to counter the man-in-the middle attack. After available of keys and tokens, the user may request for data to CSP. CSP initiates modified D-H key exchange with the user, if request is authentic. We assume that the session key (KS) is shared between CSP and the user by modified Diffie-Hellman algorithm. Now, CSP encrypts the encrypted File (Fi) and its digest (Di) with the shared session key (KS) and sends it to the user. This over encryption ensures the confidentiality of the message between cloud service provider and the user. The user then decrypts the message (user decrypts the message according to algorithm 4) and calculates the digest of File and then matches it with stored digest. If digest matches, File is original otherwise File is modified by outsiders and user then sends an error notification message to DO.

III.Algorithm

AES Algorithm(Code for Key Expansion)

KeyExpansion(byte key[4*Nk], word w[Nb*(Nr+1)], Nk)

Begin

word temp

```
i = 0
 while (i < Nk)
 w[i] = word(key[4*i], key[4*i+1], key[4*i+2], key[4*i+3])
 i = i+1
 end while
i = Nk
 while (i < Nb * (Nr+1)]
 temp = w[i-1]
 if (i mod Nk = 0)
 temp = SubWord(RotWord(temp)) xor Rcon[i/Nk]
 else if (Nk > 6 and i mod Nk = 4)
 temp = SubWord(temp)
 end if
 w[i] = w[i-Nk] xor temp
 i = i + 1
 end while
 end
```

step2: Code for the Inverse Cipher

```
InvCipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
 begin
 byte state[4,Nb]
 state = in
 AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1]) // See Sec. 5.1.4
 for round = Nr-1 step -1 downto 1
 InvShiftRows(state) // See Sec. 5.3.1
 InvSubBytes(state) // See Sec. 5.3.2
 AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
 InvMixColumns(state) // See Sec. 5.3.3
 end for
 InvShiftRows(state)
```

InvSubBytes(state)

AddRoundKey(state, w[0, Nb-1])

out = state

end

RSA Algorithm:

The RSA algorithm involves three steps: key generation, encryption and decryption. A.Key generation

RSA involves a public key and a private key. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key. The keys for the RSA algorithm are generated the following way:

1. Choose two distinct prime numbers p and q.

• For security purposes, the integers p and q should be chosen at random, and should be of similar bit-length. Prime integers can be efficiently found using a primality test. 2. Compute n = pq.

• n is used as the modulus for both the public and private keys 3. Compute $\varphi(n) = (p - 1)(q - 1)$, where φ is Euler's totient function. 4. Choose an integer e such that $1 < e < \varphi(n)$ and greatest common divisor of $(e, \varphi(n)) = 1$; i.e., e and $\varphi(n)$ are coprime.

• e is released as the public key exponent. • e having a short bit-length and small Hamming weight results in more efficient encryption - most commonly 0x10001 = 65,537. However, small values of e (such as 3) have been shown to be less secure in some settings.

B.Encryption: Alice transmits her public key to Bob and keeps the private key secret. Bob then wishes to send message M to Alice. He first turns M into an integer m, such that by using an agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext corresponding to.This can be done quickly using the method of exponentiation by squaring. Bob then transmits to Alice. Note that at least nine values of m could yield a ciphertext c equal to m,but this is very unlikely to occur in practice.

C. Decryption: Alice can recover from by using her private key exponent via computing. Given , she can recover the original message M by reversing the padding scheme.(In practice, there are more efficient methods of calculating using the pre computed values below.) r

SHA Algorithm:

Step 0: Initialize some variables

Step 1: Pick a string

Step 2: Break it into characters

Step 3: Convert characters to ASCII codes

Step 4: Convert numbers into binary

Step 5: Add '1' to the end

Step 6: Append '0's' to the end

Step 6.1: Append original message length

Step 7: 'Chunk' the message

Step 8: Break the 'Chunk' into 'Words'

Step 9: 'Extend' into 80 words

Step 9.1: XOR

Step 9.2: Left rotate

Step 10: Initialize some variables

Step 11: The main loop

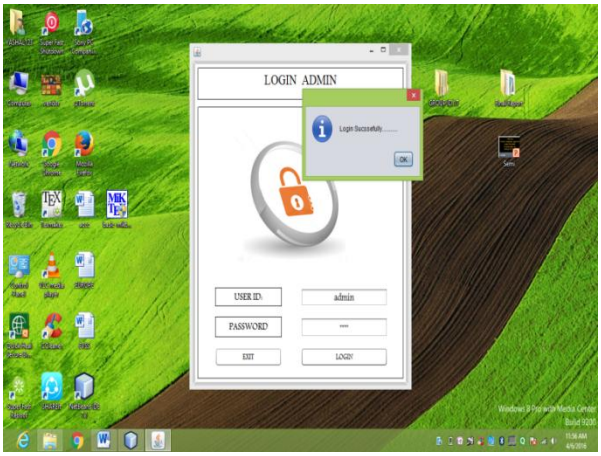
Step 11.1: Four choices

DHKE Algorithm:

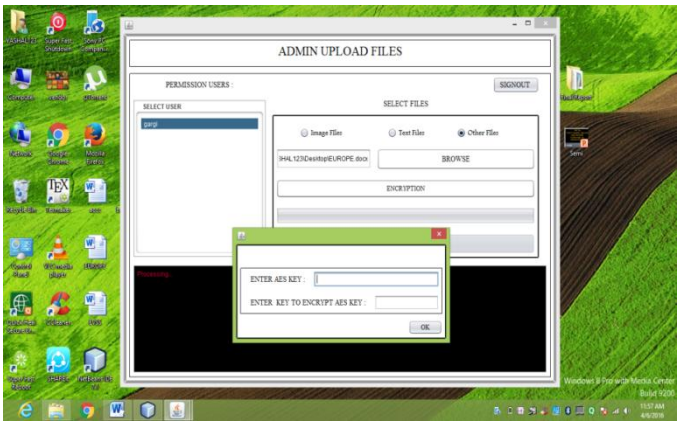
For example, Alice, Bob, and Carol could participate in a Diffie-Hellman agreement as follows, with all operations taken to be modulo :1. The parties agree on the algorithm parameters and . 2. The parties generate their private keys, named , , and . 3. Alice computes and sends it to Bob. 4. Bob computes and sends it to Carol. 5. Carol computes and uses it as her secret. 6. Bob computes and sends it to Carol. 7. Carol computes and sends it to Alice. 8. Alice computes and uses it as her secret. 9. Carol computes and sends it to Alice. 10. Alice computes and sends it to Bob. 11. Bob computes and uses it as his secret.

IV. RESULT

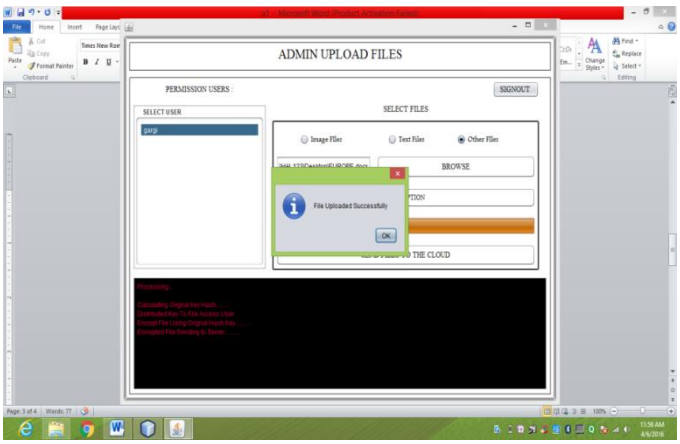
Admin Module: This is the admin login page. Admin enters the Server IP Address to login. Then Clicks on Proceed to enter the system.By executing project from Net Beans IDE7.1.



Verification of user id and password is done.

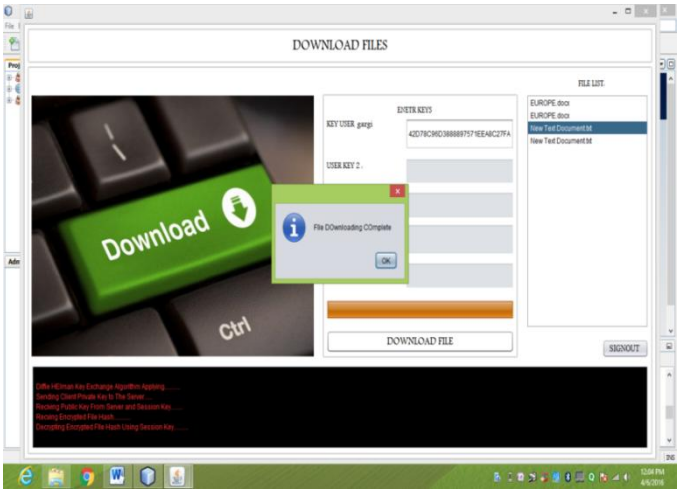


Upload files by selecting user and then by performing encryption two times the file is being sent to cloud.

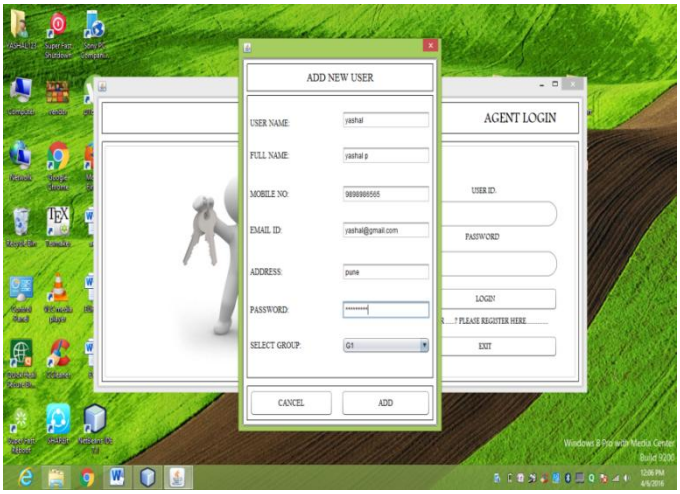


The file is uploaded successfully.

This is client portal.



In this module, the user is selected and then the key is entered then the downloading of file is performed.



In this module, new user registration is done.

V.CONCLUSION

We presented a new approach which provides security for data out sourced at CSP. Some approaches are given to secure outsourced data but they are suffering from having large number of keys and collusion attack. By employing the threshold

cryptography at the user side, we protect outsourced data from collusion attack. Since, DO stores its data at CSP in encrypted form and, keys are known only to DO and respected users group, data confidentiality is ensured. To ensure fine-grained access control of outsourced data, the scheme has used capability list. Public key cryptography and MD5 ensure the entity authentication and data integrity respectively. Public key cryptography and D-H exchange protected the data from outsiders in our approach. No of keys (because in threshold cryptography, there is a single key corresponding to each group)have reduced in the proposed scheme.

vi. References

[1] Vol.2 Issue III, March 2014 ISSN: 2321-9653 INTERNATIONAL JOURNAL FOR RESEARCH IN APPLIED SCIENCE AND ENGINEERINGTECHNOLOGY(IJRASET),Performance study on Diffie Hellman Key Exchange AlgorithmAssociate Professor, Department of Software Engineering & IT[PG] A.V.C College of Engineering, Mayiladuthurai, Assistant Professor & Head, Department of Science & Humanities Kingston Engineering College, Katpadi, Vellore.

[2] Vishal Garg et al ,Int.J.Computer Technology &Applications,Vol 3 (4), 1327-1331 Improved Diffie-Hellman Algorithm for Network Security Enhancement ,Vishal Garg ,Assistant Professor ,Department of Computer Science and Engg., JMIT Radaur, Yamunanagar, Haryana, India ,Rishu ,Department of Computer Science and Engg., JMIT Radaur, Yamunanagar, Haryana, India ISSN: 2319-5967 ISO 9001:2008 Certified International Journal of Engineering Science and Innovative Technology (IJESIT) Volume 1, Issue 2, November 2012 Diffie-Hellman and Its Application in Security Protocols,MaryamAhmed,BaharanSanjabi,DifoAldiaz.

[3] C. Boyd, W. Mao, and K. G. Paterson, Key agreement using statically keyed authenticators, in Proc. ACNS 2004, pp.248262.

[4] Miss. Pooja P. Taral1, Prof. Vijay B. Gadicha2 CSE Department, PRPCOET, SantGadge Baba Amravati University, India 2CSE Department, PRPCOET, SantGadge BabaAmravati University, in Secure Key Exchange over Internet,India 1 taralpooja@gmail.com; 2 vgadicha@rediffmail.com.

[5] Mrs. S. Saranya1 , R. Pitchandi2 1A.P(OG), 2M.Tech(CSE) Scholar Department of Computer Science and Engineering, SRM University, Chennai, India, in Authenticated Deniable Internet Key Exchange .