Scientific Journal Impact Factor (SJIF): 1.711

e-ISSN: 2349-9745
p-ISSN: 2393-8161

International Journal of Modern Trends in Engineering and Research
www.ijmter.com

Authenticating Secret Key-Exchange Between Two Parties

Dinesh Raj.R[1], Saravana Kumar.S[2], Saravanan.N[3]

[1,3] *Department of ME-CSE, Srinivasan Engineering College*
[2] *Assistant Professor/CSE, Srinivasan Engineering College*

Abstract— Key-exchange, in particular Diffie–Hellman key exchange (DHKE), is among the core cryptographic mechanisms for ensuring network security. For key-exchange, both security and privacy are desired. The proposed scheme is used to exchange the secret key by implementing the Shamir secret sharing scheme. The secret can be splitted into shares based on the threshold values, it refers to way for distributing a secret amongst a discussion of sender and receiver and the other parties include in transaction. The secret key will be reconstructed solely through equal number of shares. Individual shares are of no use on their own. Using Shamir Secret sharing, we have been enhance the approach for this present work, that it will reconstruct the secret key through the procedure of dividing into parts that is given into the each specified participants on own unique part, and some of the parts are need that to be reconstruct the secret key only the receiver verified based on their unique identity. Then the receiver receives the key of all the parts and decryption is made to message. In this work, introduce an new approach of sharing message in the way of authentication, it will produce more authentication than other secret sharing methods, because the reconstruction of key is determined and the message authentication is undetermined to hackers or illegal users.
Keywords — Authentication, Key-exchange, network security, Shamir secret sharing.

I. INTRODUCTION

Secret sending of Key, refers to ways that for distributing a secret amongst a discussion of sender and receiver and therefore the different parties embrace in dealings, whom is assigned a share of the key, the key are going to be reconstructed entirely through snug selection with the splitted recursions, of presumptively different types, of shares square measure combined together; individual shares square measure of no use on their own. victimization Shamir Secret Key sharing, we've been enhance the approach for this gift work, that it'll reconstruct the key through the procedure of dividing into elements that's given into the every such halficipants on own distinctive part, and a few of the elements square measure would like that to be reconstruct whereas it receives to Associate in Nursing receiver of the message. Then the receiver receives the key of all the elements and secret writing is formed to message. During the work, we've introduce Associate in Nursing new approach of sharing message within the means of authentication, it 'llmanufacture a lot of authentication than different secret sharing sharing strategies, as a result of the reconstruction of key's determined and therefore the message authentication is undetermined to hackers or felonious users.

The security is the main goal of this proposed scheme. The secrete Key is generated by the segmentation process. The sender sends the encrypted cypher-text to the Admin and the admin sends the data to the receiver. Here the keys are segmented for the strong security in the network. The Secrete key which are created by the sender is segmented and it send to the Admin. The Key Manager is the new part which are employed for strong security in the network. The Admin Send the key to the key manager and after the analysis it reaches the receiver and the process of decryption.

Ensure the security by using the (DHKE) core cryptographic mechanism. Deniable Internet Key-Exchange (DIKE) protocol is to provide the PKI, Identity based setting and provide useful privacy protection. The IKE protocols to ensure Internet security, which specify KE mechanism used to establish shared key for use in the IP sec standards. IKE and Ip sec can offer confidentiality, authentication and privacy. However, the beauty of using deniable key-exchange is that if the key-exchange protocol is deniable, then all the transactions (of public messages) using the session key produced by the key-exchange protocol can be deniable for both the protocol participants. Moreover, for the IKE protocol that is the core cryptographic protocol to ensure Internet security, offering deniability by IKE running at the I Player within the IPSec standard is much more desirable, because it enables various privacy services to be offered at the higher layers with uncompromised quality. Note that a privacy problem at the IP layer can cause irreparable privacy damage at the application layer. For example, an identity connected to an IP address, if not deniable, certainly nullifies an anonymous property offered by a fancy cryptographic protocol running at the application level. (If deniability is not desired, foursome cases, then a non-reputable proof, e.g., a signature, can always be issued at the application level.)Despite its seemingly conceptual simplicity, designing "sound" and "right" key-exchange protocols turns out to be extremely error prone and can be notoriously subtle. Also, the analysis of even a simple cryptographic protocol in intricate adversarial settings like the Internet can be a luxury and dauntingly complex task.

II. RELATED WORK

First login the user. If the user is new user register first then login. Existing user enter user id and password to login the network. The receiver is request the file from sender via admin. Sender selects the requested file. The secret key is generated by the key manager. Based on the secret key the file can be encrypted. The encrypted file is send to the receiver through admin. The sender can divide the secret key into shares. Pieces are give it to the key managers though admin. But the receiver may not decrypt the file without the secret key. Receiver request the secret key to admin and then the admin verifies the receiver by his/her unique identity, then only key manager provide key pieces to receiver via admin. The receiver collect all the pieces to merge the key pieces to form the secret key to decrypt the file.

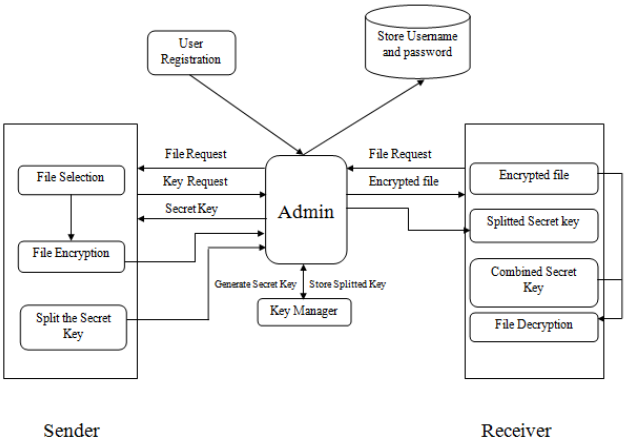

Figure 1 system architecture

Ensure the security by using the (DHKE)[1] core cryptographic mechanism. Deniable Internet Key-Exchange (DIKE) protocol is to provide the PKI, Identity based setting and provide useful privacy protection. The IKE protocols to ensure Internet security, which specify KE mechanism used to establish shared key for use in the IP sec standards. IKE and Ip sec can offer confidentiality, authentication and privacy. However, the beauty of using deniable key-exchange is

that if the key-exchange protocol is deniable, then all the transactions (of public messages) using the session key produced by the key-exchange protocol can be deniable for both the protocol participants. Moreover, for the IKE protocol that is the core cryptographic protocol to ensure Internet security, offering deniability by IKE running at the I Player within the IPSec standard is much more desirable, because it enables various privacy services to be offered at the higher layers with uncompromised quality. Note that a privacy problem at the IP layer can cause irreparable privacy damage at the application layer. For example, an identity connected to an IP address, if not deniable, certainly nullifies an anonymous property offered by a fancy cryptographic protocol running at the application level. (If deniability is not desired, foursome cases, then a non-reputable proof, e.g., a signature, can always be issued at the application level.)Despite its seemingly conceptual simplicity, designing "sound" and "right" key-exchange protocols turns out to be extremely error prone and can be notoriously subtle. Also, the analysis of even a simple cryptographic protocol in intricate adversarial settings like the Internet can be a luxury and dauntingly complex task.

III. SCOPE OF THE PAPER

This proposed work is mainly focused and introduce a new approach of sharing message in the way of authentication, it will produce more authentication than other secret sharing methods, because the reconstruction of key is determined and the message authentication is undetermined to hackers or illegal users. Secret splitting of Key, refers to ways for distributing a secret amongst a discussion of sender and receiver and the other parties include in transaction, whom is allotted a share of the key, the secret key will be reconstructed solely through comfortable variety with the splitted recursions or based on their equal number of threshold values, of presumably differing kinds, of shares are combined together; individual shares are of no use on their own. Using Shamir Secret Key sharing, we have been enhance the approach for this present work, that it will reconstruct the secret key through the procedure of dividing into parts that is given into the each specified participants on own unique part, and some of the parts are need that to be reconstruct while it receives to an receiver of the message. The secret key has to be generated by the key manager to get the authority to generate the keys and addition to that the secret key is splitted by using the Shamir's secret sharing scheme.

The receiver requests the file from administrator or server the admin to know the original file owner. The original file is selected from the owner then the owner or sender request to admin to get the secret key to encrypt the original file. Then the file owner encrypt the file and to use the secret key have been splitted based on the Shamir secret scheme based on the threshold values. Then the key pieces are to be stored in to multiple key managers. The encrypted file is passed to requested receiver then the receiver request the secret key.

Admin verifies the receiver identity then only to provide the secret key pieces.Then the receiver receives the key of all the parts and decryption is made to message. Shamir's secret sharing is an algorithm that divides a secret into shares. Secret can be recovered by combining certain numbers of shares. Imagine a case where you have to encrypt some data. No matter which encryption method you use, you must store the secret key used in the encryption in order to decrypt later. The key has to be very secured. If the key is stolen by attacker, your data will be easily decrypted. However, storing key is always difficult problem. It gets even more difficult if you need to share the key with others. However, if you use Shamir's secret sharing algorithm, you can solve the two problems to greater extent.

Shamir's secret sharing

As the name implies, Shamir's secret sharing is created by Adi Shamir. Shamir's secret sharing is an algorithm that split a secret into shares. Secret can be recovered by combining certain numbers of shares. Imagine a case where you have to encrypt some data. No matter which encryption method you use, you must store the secret key used in the encryption in order to decrypt later.

The key has to be very secured. If the key is stolen by attacker, the encrypted data will be easily decrypted. However, storing key is always difficult problem. It gets even more difficult if you need to share the key with others. This problem of storing and sharing secret key is cause of headache for administrators.

However, if you use Shamir's secret sharing algorithm, you can solve the two problems to greater extent. You can split the secret key into pieces and distribute them to other administrators. Each administrator still needs to keep a piece of secret key, but knowing a piece is not enough to recover the original secret.

Because attacker must compromise multiple administrator's pieces, secret generated by Shamir's secret sharing is very difficult to be compromised.

```
# First, you need to instantiate ShamirSecret class.
# You can specify the number of threshold in the first argument. In this case, two shares are required.
# You can pass a message to encrypt in the second argument.
shamirsecret = ShamirSecret.new(2, "In the name of Adi Shamir")

# We compute shares from the given secret. Let's assume we want to distribute to three parties, so lets create three shares.
# The argument is so called share number. You will know what it is later in this post.
# For now, just remember that it has to be unique number.
s1 = shamirsecret.compute_share(1)
s2 = shamirsecret.compute_share(2)
s3 = shamirsecret.compute_share(3)

# Once we computed shares, we will throw the secret away because we should be able to recover from shares.
shamirsecret = nil

# Then we will recover the secret. Instantiate ShamirSecret again. We don't pass secret this time because we just want to recover secret.
shamirsecret = ShamirSecret.new(2)

# Now we can recover the secret by giving two shares or more since we set threshold to be 2.
shamirsecret.recover_secretdata([s1,s3])
=> "In the name of Adi Shamir"
```

IV. CONCLSION

In this work, Develop a secret key exchange mechanism by using the Shamir secret sharing algorithm.(1)Secret: Secret is a secret message or number that you want to share with others securely.(2) Share: Share is a piece of secret. Secret is divided into pieces and each piece is called share. It is computed from given secret. In order to recover the secret, you need to get certain numbers of shares.(3)Threshold: Threshold is the number of shares you need at least in order to

recover your secret. You can restore your secret only when you have more than or equal to the number of threshold.(4) Merge : After collecting all the shares to merge the shares and to get the original secret key to decrypt the file to view the original file.

REFERENCES

[1] S. Al-Riyami and K. Paterson, "Certificateless public-key cryptography," in Proc. Asiacrypt 2003, pp. 452–473.
[2] M. Bellare and P. Rogaway, "Entity authentication and key distribution,"in Proc. CRYPTO 1993, pp. 273–289.
[3] M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," in Proc. ACM CCS 1993, pp. 62–73.
[4] D. Boneh and M. Franklin, "Identity-based encryption from the weilpairing," in Proc. CRYPTO 2001, pp. 213–229.
[5] C. Boyd, W. Mao, and K. G. Paterson, "Deniable authenticated key establishment for Internet protocols," in Proc. SPW 2003, pp. 255–271.
[6] C. Boyd, W. Mao, and K. G. Paterson, "Key agreement using statically keyed authenticators," in Proc. ACNS 2004, pp. 248–262.
[7] C. Boyd and A. Mathuria, Protocols for Authentication and Key Establishment. New York, NY, USA: Springer-Verlag, 2003.
[8] C. Brzuska, N. P. Smart, B. Warinschi, and G. J. Watson, "An analysis of the EMV channel establishment protocol," in Proc. ACM CCS, 2013,pp. 373–386.
[9] J. Camenisch, N. Casati, T. Gross, and V. Shoup, "Credential authenticated identification and key exchange," in Proc. CRYPTO 2010,pp. 255–276.
[10] R. Canetti, "Security and composition of cryptographic protocols: A tutorial," SIGACT News, vol. 37, no. 3, pp. 67–92, 2006.
[11] R. Canetti, U. Feige, O. Goldreich, and M. Naor, "Adaptively secure multi-party computation," in Proc. STOC 1996, pp. 639–648.
[12] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in Proc. Eurocrypt 2001,pp. 289–307.
[13] R. Canetti and H. Krawczyk, "Security analysis of IKE's signature-based key-exchange protocol," in Proc. CRYPTO 2002, pp. 143–161.
[14] K. K. Choo, C. Boyd, Y. Hitchcock, and G. Maitland, "On session identifiers in provably secure protocols," in Proc. SCN 2004,pp. 351–366.
[15] C. J. F. Cremers, "Formally and practically relating the CK, CK-HMQV,and eCK security models for authenticated key exchange," IACR (The International Association for Cryptologic Research), San Diego, CA,USA, Tech. Rep. 2009/253, 2009.
[16] I. Damgård, "Towards practical public-key systems secure against chosenciphertext attacks," in Proc. CRYPTO 1991, pp. 445–456.
[17] Y. Dodis, J. Katz, A. Smith, and S. Walfish, "Composability and on-linedeniability of authentication," in Proc. TCC 2009, pp. 146–162.
[18] C. Dwork, M. Naor, and A. Sahai, "Concurrent zero-knowledge," inProc. STOC 1998, pp. 409–418.
[19] M. C. Gorantla, R. Gangishetti, and A. Saxena, "A survey on IDbased cryptographic primitives," IACR (The International Association for Cryptologic Research), San Diego, CA, USA, Tech. Rep. 2005/094,2005

