International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 05 | May - 2021 ISSN: 2582-3930

A STUDY ON

Diffie - Hellman Key exchange secure text transfer based on Cloud

Aman Dongre[1], Sakshi Band[2] , Pranay Beherkhede[3],Sonal Bopulkar[4] , Purva Kantode[5]

B.E IV Year Students, Department of Electronics and Telecommunication
Vivek Deshmukh[6]
Assistant Professor
S B Jain Institute of Technology Management and Research, Nagpur, Maharashtra, India

Abstract : Encryption is the procedure of concealing private or touchy data inside something that has all the earmarks of being nothing be a standard thing. On the off chance that an individual perspective that figure text, the person will have no clue about that there is any privileged data. What encryption basically does is misuse human insight, human faculties are not prepared to search for records that have data within them. What this framework does is, it lets client to send message as emit message and gives a key or a secret key to bolt the content, what this key does is, it scrambles the content, so that regardless of whether it is hacked by programmer it won't peruse the content. Beneficiary will require the way to decode the secret content. Client at that point sends the way in to the collector and afterward he enters the key or secret phrase for unscrambling of text, he at that point presses decode key to get secret content from the sender. Diffie-Hellman key trade offers the best of both as it utilizes public key procedures to permit the trading of a private encryption key. By utilizing this technique, you can twofold guarantee that your secret message is sent subtly without outside obstruction of programmers or saltines. On the off chance that sender sends this code text in open public others won't realize what is it, and it will be gotten by collector. The framework utilizes online data set to store all connected data. As, the venture documents and a data set record will be put away into the Azure cloud, the task will be gotten to in the internet browser through Azure connection.

Keywords : Azure connection, Encryption.

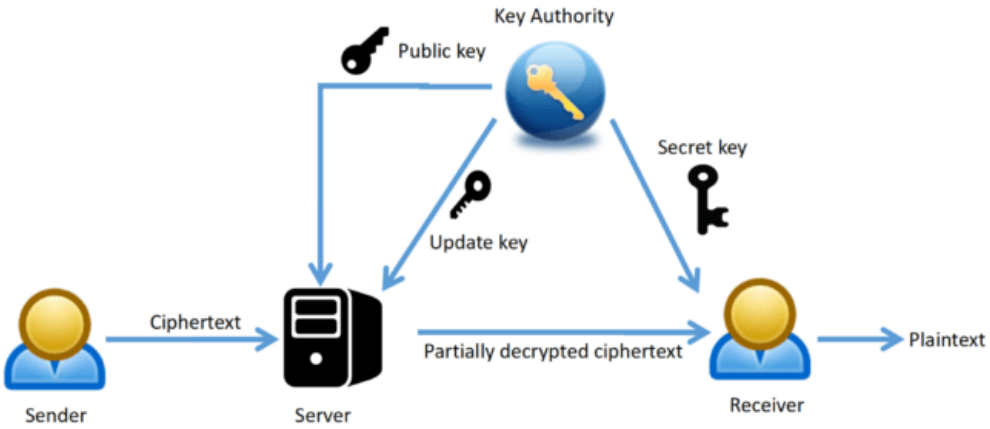
Introduction : Current cryptography is utilized in Computer and Communication. Current cryptography utilizes binary bit succession. Investigation of cryptosystem is called Cryptology. Cryptography and Cryptanalysis are the two parts of Cryptology. Breaking and getting the data part goes under Cryptanalysis. Cryptography gives data security by giving Confidentiality, Data Integrity, Authentication, furthermore, Non-renouncement to administrations. A cryptosystem is the execution of cryptographic methods. Acryptosystem is comprised of plain content, encryption calculation, figure text, decoding calculation, theencryption key and decoding key. There are two essential sorts of a cryptosystem, symmetric keycryptosystem and hilter kilter key cryptosystem.

In Symmetric key cryptosystem, a similar key is utilized for encryption and unscrambling. Key foundation and Trust Issue are the two principle challenges in symmetric key cryptosystem. In Asymmetric key cryptosystem, diverse key is utilized for encryption and decoding. Here, the keys are numerically identified with one another. Uneven key cryptosystem is additionally called as Public key cryptosystem. The Public-key cryptography comes in the image alongside the Diffie-Hellman calculation. Public-Key Cryptography implies just Asymmetric Cryptography which utilizes public and private keys to encode and decode the message. This idea was utilized by Diffie and Hellman to trade the mysterious key for sending and getting the messages. Quite possibly the most basic issue in cryptography is trading the key between two conveying gadgets. It was not about building up a common mystery key, yet it was going to do it so that any individual who is there at the correspondence between the gadgets don't discover the key. Diffie-Hellman calculation was first loan boss was Ralph Merkle and this calculation is named after Whitfield Diffie and Martin Hellman. This calculation makes the key trade secure over a public channel. The Diffie – Hellman is utilized for public key cryptography, SSL, SSH, PGP and other PKI frameworks. Many web administrations employments

Diffie–Hellman for solid correspondence and for getting purpose. The most astonishing thing inDiffie-Hellman key trade is the correspondence among sender and collector will occur over the public channel and for assailant currently it's getting conceivable. A few assaults which is conceivable on the Diffie-Hellman calculation: man-in-

centreattack, plain-text assault, logjam assault, and so on Logjam assault isanother kind of assault found on the Diffie-Hellman key-trade convention which is utilized in TLS. We have proposed a redesigned Diffie-Hellman calculation for safer and solid key trade also, for solid data trade between the sender and the recipient.

The Diffie-Hellman Algorithm : The Diffie-Hellman key trade is the most generally utilized technique for creating and trading keys by means of an unreliable channel. The Diffie–Hellman key trade makes a common mystery between the two gatherings which is utilized for secret correspondence for trading the information over open organizations. Assuming you need to trade some restricted intel to someone else, the best method to do so is scrambling the message with a code and coordinate the sort of code and key that you are wanting to use ahead of time or over a protected correspondence channel. Say, you are communicating something specific encoded with shift-figure in which each 'a' becomes 'b' and 'b' becomes 'c, etc. So let the message be 'hub is lovely' which becomes 'opef jt cfbvjgvm'. Assume the code is uncracked by the Man in the Middle and is securely gotten by the following individual. Be that as it may, imagine a scenario in which you were unable to mastermind code ahead of time with the collector. Despite the fact that the message will be adequately secure to be decoded by the Man in the Middle yet additionally the recipient will not have the option to interpret. This issue was settled by the Diffie-Hellman key trade. The Diffie-Hellman key trade calculation permits the individuals who have never met ahead of time to make a common key wellbeing, even though an unreliable channel.



Algorithm :

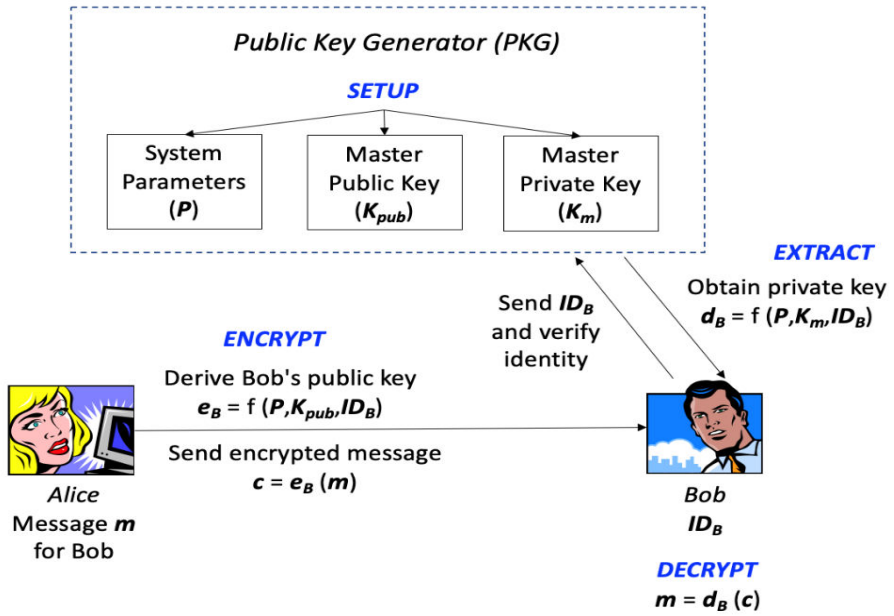
The Diffie-Hellman Key Exchange algorithm involves the following steps:

- The sender and receiverpick an indivisible number p and itsprimitive root g.
- Then the two of them pick a private key each let it be an and b, this private is simply known to them.
- The public key of both the sender and beneficiary is determined for sender it is equal to

  $A = g^a \text{mod} p$and for receive it'$sB = g^b \text{mod} p$.

  where A and B address the particular public key.

- The public keys produced are then traded by both the sender and receiver.
- Now both the sender and receiver compute their mysterious key i.e. sender's secret key is be gotten by
  $S = B^a \text{mod} p$.
  what's more, correspondingly for collector
  $S = A^b \text{mod} p$.
- where S addresses the mysterious key produced.
- This 'S' is the common secret key which can be utilized in symmetric calculations for encryption and decryption.



Example for the Algorithm :

Step 1: Alice and Bob get public numbers P = 23, G = 9

Step 2: Alice selected a private key a = 4 and

Bob selected a private key b = 3

Step 3: Alice and Bob compute public values

Alice: x =(9^4 mod 23) = (6561 mod 23) = 6

Bob: y = (9^3 mod 23) = (729 mod 23) = 16

Step 4: Alice and Bob exchange public numbers

Step 5: Alice receives public key y =16 and

Bob receives public key x = 6

Step 6: Alice and Bob compute symmetric keys

Alice: ka = y^a mod p = 65536 mod 23 = 9

Bob: kb = x^b mod p = 216 mod 23 = 9

Step 7: 9 is the shared secret

.

Technological stacks :

- **Operating system used :**

  Windows 10 and Android version compatible with 8.0 and above.

- **Technology used :**

  Cloud server is used for running microservices.

  Android app for both sender n receiver.

  Cloud server helps with real time authentication in communication with sender n receiver also for android receiver.

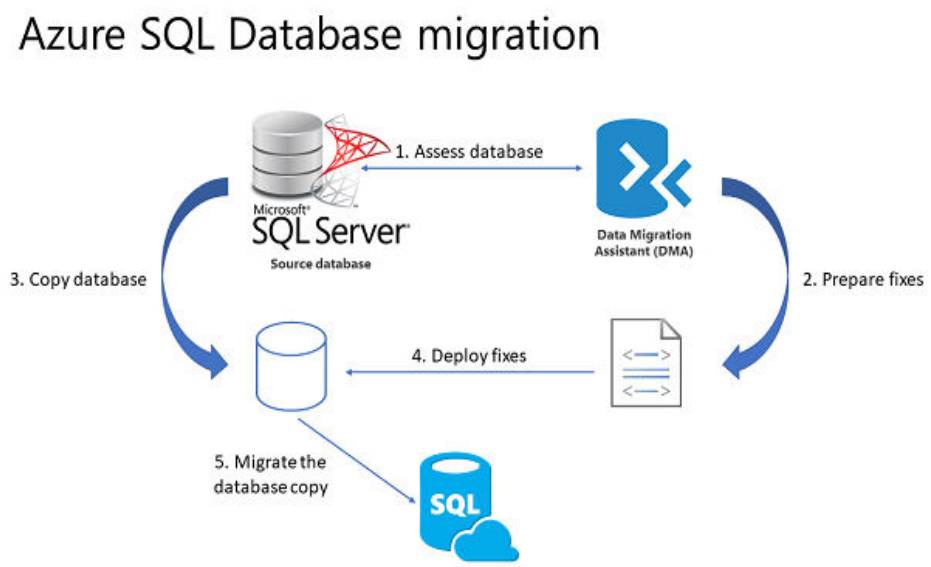
- **Preferred Languages :**

Java, Xml, Sql

Java is used as the server-side language for most back-end development projects, including those involving big data and Android development.

XML (Extensible Markup Language) is an extremely famous straightforward book-based language that can be utilized as a method of correspondence between various applications. It is considered as a standard way to ship and store information.

SQL is a standard **language** for storing, manipulating and retrieving data in databases.



Framework : Hibernate and Springboot.

Hibernate ORM is an object–relational mapping tool for the Java programming language. It provides a framework for mapping an object-oriented domain model to a relational database.

InSpring boot microservices will be deployed in cloud server. The primary objective of the Spring Boot structure is to decrease overall development time and increase effectiveness by having a default arrangement for unit and integration tests.

- **Software for Development :**

- Eclipse IDE for s/w dev.
- MySQL client for db. handling.
- Android Studio IDE for mobile app dev.

- **Design Pattern Used :**

Architecture :-

Microservices and REST Api :

MVC (Model View Controller) :-

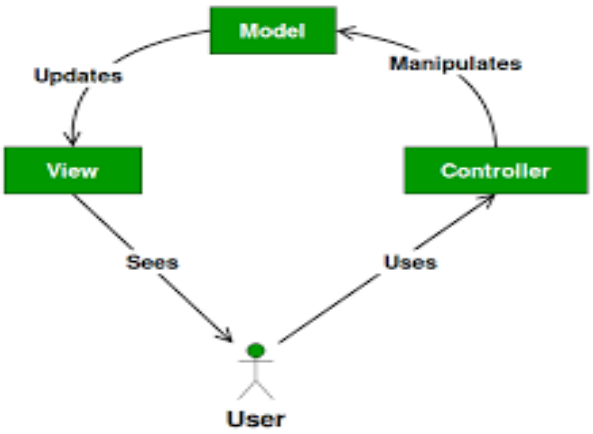
Model : It is the focal part of the pattern. It is the application's dynamic data structure, autonomous of the UI. It straightforwardly deals with thedata, logic and rules of the application.

View : Any representation of information, chart or table. Numerous perspectives on a similar data are conceivable, for example, a bar outline for the board and tabular view for accountants.

Controller : Acknowledges information and converts it to orders for the model or view. As well as partitioning the application into these segments, the model–view controller design defines the interactions between them.The model

is liable for dealing with the information of the application. It gets client contribution from the regulator. The view renders show of the model in a specific organization. The regulator reacts to the client input and performs connections on the information model articles. The regulator gets the info, alternatively approves it and afterward passes the contribution to the model.

Likewise with other programming designs, MVC communicates the "core of the solution" to a problem while permitting it to be adjusted for every framework. Specific MVC plans can fluctuate altogether from the customary description here.



Although initially produced for work area processing, MVC has been generally embraced as a plan for World Wide Web applications in significant programming dialects. A few web systems have been made that uphold the example. These software frameworks vary in their interpretations, mainly in the way that the MVC responsibilities are divided between the client and server.

Some web MVC systems adopt a slender customer strategy that places practically the whole model, view and regulator rationale on the worker. In this methodology, the customer sends either hyperlink demands or structure entries to the regulator and afterward gets a total and refreshed website page (or other report) from the view; the model exists altogether on the worker.

The Model View Controller (MVC) plan design indicates that an application comprises of a data model, show data, and control data. The example necessitates that each of these be isolated into different objects.

MVC is a greater amount of an architectural pattern, however not for complete application. MVC for the most part identifies with the UI/connection layer of an application.

- **Project Flow Design :**

Sender app will send request to spring boot application which contain microservices which will be used for accepting request from android app and receiver app basically needed for authentication and communication after taking req from android application it will authenticate the receiver by using Diffie-Hellmanalgorithm which will internally pass the secret keys andafter successful authentication of the sender it will send positive acknowledgment to sender.

In between the receiver application will receive the message from sender application if and only if the sender application is genuine and successfully authenticated.No communication or session can establish if the authentication is successful between sender app and receiver spring boot microservices is mainly responsible for this authentication. It will not only authenticate the app but it will store the messages in database using ORM tool call hibernate.

This is all above technologicalperspective and flow.

Result and Analysis :

Man-in-the-center attack and is finished by keeping the public keys of sender and collector by the attacker. Also, sending counterfeit public keys to them individually. Here, the improved Diffie-Hellman calculation gets the key as it's creating the second common secret key and attacker is uninformed of the possibility of taking the primitive base of the main secret key. The known-plaintext attack utilizes plaintext and cipher text for recovering the secret keys. Thus, this attack is a lot of conceivable on Diffie-Hellman algorithm. So, Known-plaintext attack is perhaps the most plausible attack in the first DiffieHellman calculation. Yet, in the overhauled Diffie-Hellman algorithm, the common secret key is being created for the second time for example second shared-secret key and attacker here is totally unaware of

the thought, we are utilizing of taking 'e' as the crude base of first shared-secret key 'S' and afterward multiplying individual arbitrary number to the second common secret key and afterward trading it for creating key each an ideal opportunity for each message or in any event, for a similar message. Thus, this will make our framework especially secured .

Security of the message gets influenced because of this attack on the grounds that the attacker who is in the sender and beneficiary of the message alters the message and alters it as per his/her need. There are some notable arrangements accessible for the anticipation of this attack. Our point is to survey those accessible methods The execution season of the upgraded algorithm is a lot more prominent than the first Diffie-Hellman calculation. The contrast between the two runtimes is very small. So, it is feasible to present second-shared secret key and arbitrary boundaries in the first Diffie-Hellman algorithm.

Conclusion : In this way, when all is said in done, we've tracked down that the Diffie-Hellman calculation makes it simpler for trading the mysterious key produced which would then be able to be utilized in symmetric calculations like AES to play out the encryption and decoding activities. However, the overall Diffie-Hellman is inclined to a few attacks making it not so dependable for playing out the mysterious key trade. The philosophy proposed in this paper has been tried on various lengths of the content and the outcome got is discovered to be a lot engaging. The calculation is tried for different security issues and discovered to be invulnerable for the previously mentioned assaults. Subsequently we can utilize the key produced from this in the symmetric calculations like AES and make it considerably safer.

References :

1. Tirthani, Neha and Ganesan R 2014 Data Security in Cloud Architecture Based on Diffie Hellman and Elliptical Curve Cryptography IACR Cryptology e-Print Archive 49
2. Mandal, Sayonnha and AbhishekParakh 2015 Implementing Diffie-Hellman key exchange using quantum EPR pairs Proc. SPIE. 9500
3. Koziel, Brian, et al. 2016 Neon-Sidh: efficient implementation of supersingular isogeny DiffieHellman key exchange protocol on ARM International Conference on Cryptology and Network Security Springer International Publishing
4. Yao, Andrew C and Yunlei Zhao 2013 Method and structure for self-sealed joint proof-ofknowledge and Diffie Hellman key-exchange protocols U.S. Patent No. 8,464,060
5. Sheffer Y and Fluhrer S 2013 Additional Diffie-Hellman Tests for the Internet Key Exchange Protocol Version 2 (IKEv2). No. RFC 6989.
6. W. Diffie and M. E. Hellman, New directions in cryptography, IEEE Trans. Inform. Theory, Nov. 1976,22:644-654.
7. ManojRanjanMishra,andJayaprakashKar,ASTUDYONDIFFIE-HELLMANKEYEXCHANGE PROTO-COLS,inInternationalJournalofPureand Applied Mathematics, 2017, Volume 114 No. 2:179-189.
8. Diffie–Hellman keyexchange-https://www.wikiwand.com/en/DiffieE28093Hellman-key-exchange.

9. Advanced EncryptionStandard-https://www.wikiwand.com/en/Advanced-Encryption-Standard.
10. Security issues of the Diffie-Hellman key exchange-https://www.comparitech.com/blog/information-
11. C.G.Günther,AnIdentity-BasedKey-ExchangeProtocol,inAdvancesinCryptology
12. EUROCRYPT 89, Springer Berlin Heidelberg, 1990, pp. 29-37.
13. E.OkamotoandK.Tanaka,Keydistributionsystembasedonidentificationinformation, SelectedAreasinCommunications,IEEEJournalon,vol.7,no.4,pp.481-485,1989.
14. Fiat and A. Shamir, How to prove yourself: practical solutions to identification and signatureproblems,inAdvancesincryptology—CRYPTO86,Springer-VerlagLondon, 1987, pp. 186-194.
15. W. Diffie, P. C. V. Oorschot and M. J. Wiener, Authentication and authenticated key exchanges,Designs,CodesandCryptography,vol.2,no.2,pp.107-125,June1992.
16. M. Just and S. Vaudenay, Authenticated Multi-Party Key Agreement, in ASIACRYPT '96ProceedingsoftheInternationalConferenceontheTheoryandApplicationsofCryptology and Information Security: Advances in Cryptology,1996.