*Secure text transfer using Diffie-Hellman key Exchange Based on Cloud*

Abhay Tomer

Apex Institute of Technology, Chadigarh University

Mohali, Punjab

20BCS3566@cuchd.in

***Abstract*— This paper outlines a system for securely and efficiently managing data in the cloud using a public key database with strong encryption. The system allows data owners and users to share a key while keeping the rest of the data encrypted, private and disconnected. To enhance security, the framework uses advanced encryption methods such as AES and Diffie-Hellman for secure transactions and cloud security. The Diffie-Hellman key exchange allows two parties to create a shared key over a secure channel without prior communication, which can be used for secure data exchange in the cloud. This system provides a secure and scalable solution for managing data and sharing information in the cloud.**

***Keywords—Diffie Hellman,Cryptography, Key exchange, Cloud security.***

1. INTRODUCTION

The cloud computing today has become a trendy method of storing and accessing data, but concerns regarding unauthorized access by malevolent entities pose a significant security threat. the Diffie-Hellman key exchange algorithm has emerged as a viable cryptographic protocol for establishing a shared secret key between two parties over an insecure network, which can be utilized to encrypt and decrypt data, ensuring confidentiality, integrity, and authenticity of the transferred data. By utilizing the Diffie-Hellman algorithm for data transfer in the cloud, sensitive information can be securely transmitted between the client and cloud server.

The primary advantages of the Diffie-Hellman algorithm is that it provides a secure means of key exchange without the need for pre-shared keys or complex public key infrastructure (PKI). This makes it an appropriate solution for secure data transfer in the cloud, where there may be multiple users accessing the same data.

However, like any cryptographic algorithm, the Diffie-Hellman key exchange has its limitations and potential vulnerabilities that need to be addressed. These include the vulnerability to man-in-the-middle (MITM) attacks, weak passwords or key exchange parameters[2], lack of authentication, implementation flaws, and the potential threat of quantum computing.

MITM attacks can compromise the security of the key exchange by allowing an attacker to intercept communication between two parties and establish a separate key exchanges with both users and act as an intermediary. Weak key exchange parameters can make the key exchange susceptible to attacks such as brute-force or discrete logarithm attacks[1]. Lack of authentication means parties cannot verify each other's identities, making it vulnerable to impersonation or spoofing attacks[3].

Implementation flaws or vulnerabilities in the software or systems used for the key exchange can compromise the security of the exchanged keys. With the advent of quantum computing, it is possible that certain quantum algorithms could potentially break the security of Diffie-Hellman key exchange, posing a threat to the security of communications that rely on this algorithm in the future.

To mitigate these risks, appropriate security measures need to be implemented, such as using strong key exchange parameters, authenticating the users, and making sure that all your software and systems are up-to-date with all the patches. Additionally, considering alternative cryptographic algorithms, such as post-quantum cryptography, that may be resistant to quantum computing. It can be a prudent approach for future security considerations.

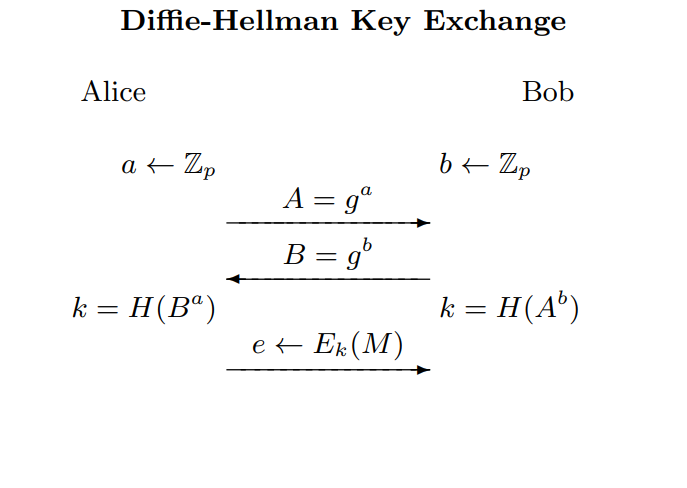
In conclusion, while the Diffie-Hellman key exchange algorithm provides a secure means of key exchange, it is not immune to vulnerabilities and limitations that need to be taken into consideration. By implementing appropriate security measures and being aware of potential risks, cloud computing users can mitigate the risks of unauthorized access and protect sensitive data from malicious entities.

1. DIFFIE-HELLMAN KEY EXCHANGE
2. INTRODUCTION

The need for a key exchange protocol to securely deliver encryption keys on a insecure channel between two parties have been a concerning issue since the dawn of cryptography in the Egypt. It emphasizes the importance of secure transmission to prevent unauthorized access or accidental disclosure of data during transmission which can be detrimental. This article explains the key agreement protocol that derives the session key from information shared by both parties and the basic transport protocol that generates a session key and sends it securely to the other party.

The Diffie-Hellman key-exchange is introduced as one of the first secure way of key exchange, which allows two parties to exchange a session secret key based on public parameters shared in the initialization stage. The article highlights the efficiency of the Diffie-Hellman protocol and its vulnerability to man-in-the-middle attacks.It also discusses recent Authentication Key Exchange (AKE) protocols, their potential attacks, and their resistance to known attacks such as the Ephemeral Key Compromise (EKC) attack also the very famous Key Compromise Impersonation (KCI) attack. Additionally, the article emphasizes the need to balance cost/efficiency and security while choosing key establishment protocols and recommends avoiding public key cryptographic methods such as RSA, ECC, and ElGamal to achieve low processing costs. The Station-to-Station (STS) AKE Protocol is presented as one of the earliest authenticated key exchange protocols. Finally, secure and efficient key exchange protocols are introduced, and their vulnerabilities are described.

1. ALGORITHM

A large prime number p

A generator value g, where g is: 1 < g < p

Alice's secret number a, where a is: 1 < a < p-1

Bob's secret number b, where b is: 1 < b < p-1

An optional hash function H() for additional security

Outputs:

A shared secret key K

Steps:

1. Alice generates a random secret number a, where a is: 1 < a < p-1
2. Bob generates a random secret number b, where b is: 1 < b < p-1
3. Alice then calculates A by A = g^a modules p
4. Bob also calculatesB by B = g^b modules p
5. Alice sends A to Bob, encrypted using a secure encryption scheme (e.g., RSA)
6. Bob sends B to Alice, encrypted using the same encryption scheme
7. Alice decrypts B using her private key and calculates K by K = B^a modules p
8. Bob decrypts A using his private key and calculates K by K = A^b modules p

Congratulations, Alice and Bob now share the secret key K

Alice and Bob can also use hash function on K by a secure hash function to produce a shared secret key K'

Alice and Bob use K' as the encryption key for their messages to each other[4].

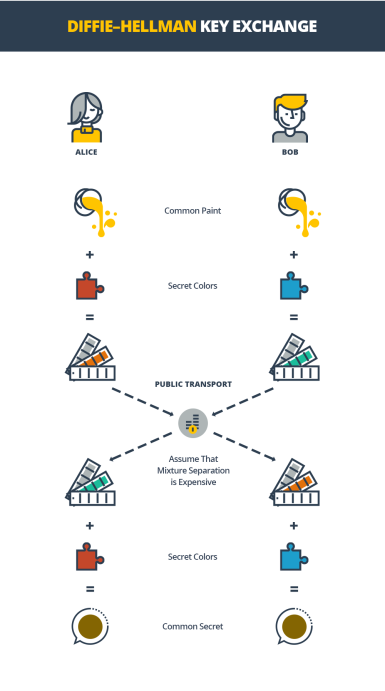


Fig 1 Representation of Diffie Hellman Key Exchange

To initiate the process, two individuals, herein referred to as Alice and Bob, must first agree upon a color. This color, while not requiring secrecy, must differ on each occasion. For the purposes of this explanation, let us assume that they have decided upon the hue of yellow. Subsequently, each party independently selects a private color which they do not disclose to the other. In our example, Alice has opted for the shade of orange, while Bob has chosen a blue-green tint. The next step involves each of them mixing their respective private colors with the jointly agreed upon color, resulting in Alice obtaining an orange-tan blend and Bob obtaining a light blue mixture. The two parties now exchange their mixed colors with each other publicly.

Finally, both parties combine the colors taken from each other with their own colors. The result is a weird yellow-brown combination that looks like Alice and Bob. If a third person intervenes and overhears the color transitions, it will be difficult to identify everyone's secret color and it will be impossible to create the final composition.In actuality, this process would entail the use of large numbers instead of colors, as computers can execute the required calculations expeditiously.

1. DRAWBACKS

Although the Diffie-Hellman key exchange mechanism is an outstanding encryption tool, the certificate procedure has a fundamental problem. It must not confuse security experts and pros. As a result, the algorithm's capacity to verify the validity of the exchange participants is called into doubt. As a result, it is subject to many forms of assaults, such as man-in-the-middle attacks, which can compromise and jeopardize the data security and integrity.

A hostile third party can intercept and manipulate communication between two parties in this form of attack. Because the technique lacks an authentication procedure, the attacker may simply mimic one or both parties and fool them into disclosing their private keys. This can jeopardise the security of the data shared and have serious implications. To counter such attacks, it is critical to employ extra safeguards such as digital certificates or digital signatures or even third party autehntication methodologies such as kerberos.

Diffie-Hellman algorithm can be resource intensive, resulting in higher resource usage and longer processing times. Due to its complex mathematical calculations, it may not be the most efficient algorithm for certain scenarios, particularly those involving limited computational resources. Additionally, the algorithm's intensive processing requirements can impact CPU performance time, potentially slowing down other important processes. Thus, while the Diffie-Hellman algorithm is a very valuable like no other tool used for secure key exchange, its high computational demands should be carefully considered in order to ensure optimal performance.[5]

Diffie-Hellman algorithm can on be used for sharing key nod for encryption of information. It only provides a secure way to exchange keys between two parties. After the key exchange, a separate encryption algorithm must be used to encrypt the actual data being transmitted. This limitation is because the Diffie-Hellman algorithm is only concerned with generating a shared secret key, but does not provide any mechanism for encrypting or decrypting data.

The Diffie Hellman calculation cannot be utilized to sign computerized marks. The Diffie-Hellman strategy can be utilized for key trades but not for advanced marks. Public-key cryptography, a on a very basic level modern cryptographic approach, is required for advanced signature procedures. Advanced marks permit you to approve the legitimacy of communications and guarantee that they are not altered with whereas in course. In spite of the fact that Diffie-Hellman may be utilized in conjunction with advanced signature methods, it cannot be used to create a advanced signature on its own. Other methods, such as RSA and DSA, are regularly utilized to create digital signature generation.

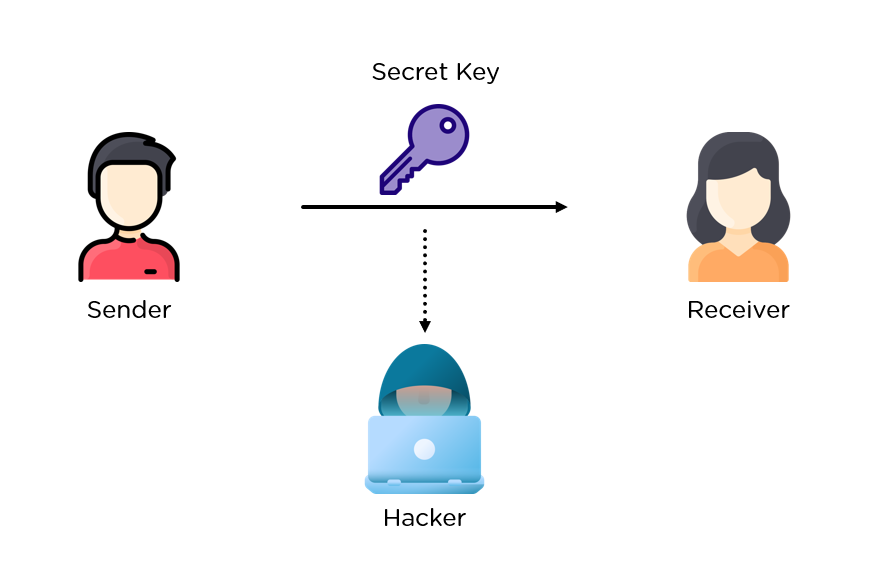


Fig 2 Man In The Middle Attack

There are several number theory-related attacks that can be launched against Diffie-Hellman key exchange such as

Small subgroup attack: This attack targets the structure used in the Diffie-Hellman key exchange, primarily admin groups. If the group has a small group (for example, a small group), the attacker can use the Pohlig-Hellman algorithm that can help solve the problem of inequality of the logarithm in group, thereby revealing the information secret. To prevent this attack, it is recommended to use groups with large leaders or without small groups, such as groups of elliptic curves.

Exponential Analysis Attack: This attack is a general technique for solving the split logarithm problem in cyclic groups. It is based on the observation that each element in a cyclic group can be expressed as generator power and the logarithm of each element can be expressed as the combination of the logarithms of the generators. This attack is more effective than the general purpose small-level large-step algorithm, especially for larger orders. To prevent such attacks, it is recommended to use large groups of leaders or groups of elliptic curves with large groups of leaders.

Quadratic remainder stop: This argument uses the fact that mod p has exactly two roots for the prime number p and the quadratic remainder a (i.e., integer squared modulo p). An attacker can use this tool to reduce logarithm inequality problems to simple quadratic problems that can be solved efficiently using the Tonelli-Shanks algorithm. To prevent this attack, it is recommended to use the prime number “p” so that “p-1” has vey large unsolvable value and makes the quadratic problem more significant. [6]

The drawbacks of the Diffie-Hellman key exchange algorithm are numerous. Lack of an authentication procedure and vulnerability to man-in-the-middle attacks make it less secure than other modern encryption methods. Furthermore, the algorithm is computationally intensive, leading to high resource and CPU performance costs. It is limited to symmetric key exchange, and encryption of information and digital signature cannot be performed using this algorithm. Therefore, while the Diffie-Hellman algorithm was groundbreaking in its time, its limitations have been exposed, and modern encryption methods have since surpassed it in terms of security and efficiency.

1. CONTEMPORARY SYSTEMS
2. INTRODUCTION

Modern cryptographic systems are utilizing the Diffie-Hellman key exchange algorithm to enable secure and efficient communication. Use of elliptic curve cryptography (ECC) in combination with Diffie-Hellman, known as ECDH, is one of the latest developments in this area[7]. This approach provides a high level of security with smaller key sizes, which leads to reduced computational and storage requirements. Another noteworthy scheme is the DHIES protocol, which combines Diffie-Hellman with symmetric encryption to ensure both data privacy and authenticity[8]. The MQV protocol, which extends the basic Diffie-Hellman key exchange to allow for mutual authentication, has also gained popularity in recent years. Furthermore, zero knowledge proofs such as ZKP-DL, are being used in conjunction with the Diffie-Hellman key exchange to authenticate participants without revealing their information about their private keys. These advancements underscore the continuing importance and relevance of the Diffie-Hellman key exchange in contemporary cryptographic systems.

1. SECURE SOCKET LAYER

SSL which is Secure Sockets Layer protocol is a widely used encryption protocol that provides secure communication on the Internet. The system is built on a frankenstein of public-key and symmetric-key cryptography, both of which have become critical aspects of cybersecurity in recent years. SSL protects communication between clients and servers by preventing access to sensitive data such as, credit card numbers, pins details, and other personal information. SSL secures the connection between client and server uses symmetric key encryption. When a client establishes a connection to a server, the server delivers the client an SSL certificate with the server's open key[9].

The client then creates a key using cryptography for the session and encrypts it with the server's public key hence, only the server can decrypt the session key. This session is then used for other communications between the client and server, allowing for fast and efficient data transfer. One of the main features of SSL is that it offers end-to-end encryption, keeping data private and tamper-proof in transit. SSL also provides authentication by allowing the client to communicate with the intended client and not a fraudster. This is done by verifying the SSL certificate presented by the server and making sure it is issued by a trusted authority.

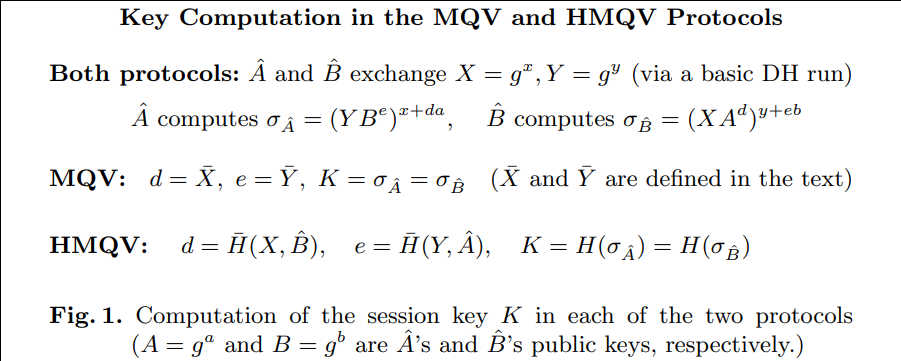
Deffi-Hellman key exchange is an essential part of SSL that allows generation of secure keys for data encryption. Using the Diffie-Hellman key exchange, SSL protects the same keys used for access by spreading across the network, making it difficult for attackers to intercept and decipher data.

1. ELLIPTIC CURVE

Elliptic Curve Diffie-Hellman also knows as ECDH is also key exchange protocol that uses the “discrete logarithm problem”[15] in the context of the new elliptic curve cryptography. ECDH is similar to traditional Diffie-Hellman, but it uses elliptic curves instead of modular exponentiation. ECDH provides equivalent security with shorter key sizes compared to traditional Diffie-Hellman. Each party in ECDH generates an elliptic curve private key that is used to calculate the corresponding public key. Public keys are exchanged by calculating secret data using each party's own private key and the other party's public key. The shared secret is used as the basis for symmetric key cryptography.

ECDH is utilized in numerous applications, counting secure communications, computerized digital certificates, and watchword administration as well as key management while sharing between servers. It is known that the Transport Layer Security (TLS) convention utilized it to secure Web associations employments between client and servers. TLS employments ECDH to set up a shared secret, which is utilized to scramble and securing the information transmitted between them.

ECDH has several advantages over traditional Diffie-Hellman. Firstly, ECDH provides equivalent security with shorter key sizes, which reduces the computational overhead of generating and transmitting keys. Secondly, ECDH is more resistant to attacks based on the discrete logarithm problem, which is more secure than traditional Diffie-Hellman. Finally, ECDH is more efficient than traditional Diffie-Hellman, making it better suited for use in applications where computational resources are limited.



Menezes-Qu-Vanstone is a key agreement protocol that is more of a Diffie-Hellman protocol extension. It allows two parties to design an safe encryption channel system, similar to Diffie-Hellman. MQV, on the other hand, offers several advantages over Diffie-Hellman, including the ability to prevent MITM attacks and use shorter public keys for faster processing. To give more security than Diffie-Hellman, MQV employs Elliptic Curve Cryptography (ECC). ECC enables the use of lower key sizes while preserving the same security intigrity as bigger key sizes in traditional cryptography. The use of shorter keys allows for faster computation, which is especially useful in resource-constrained environments such as mobile devices[10].

HMQV, A Secure Diffie-Hellman High-Performance Protocol. It is a version of conventional Diffie-Hellman key exchange designed for improving key exchange security and performance in limited contexts such as wireless sensor networks. HMQV is a high-performance and secure key exchange protocol designed for usage in resource-constrained situations. Its reciprocal authentication, forward secrecy, resilience to side-channel attacks, and rapid key exchange make it a popular choice for wireless sensor networks, mobile devices, and other resource-constrained situations.

MQV also incorporates an added layer of security through the use of a three-pass protocol. This protocol ensures that both parties can verify the authenticity of other's public keys before the connection is established. This protects against MITM, where a third party intercepts the communications and hence has the power to alter the information flowing through both end. The use of Diffie-Hellman in MQV is crucial to its key exchange process. It involves both parties agreeing on a set of parameters, including a finite field and a base point on that field. Each party then generates a public-private key pair and calculates a shared secret using the other party's public key and their own private key. This shared secret is then used to derive a symmetric key for secure communication.

Therefore, MQV provides a more secure and efficient method for key exchange than traditional Diffie-Hellman. Its use of elliptic curve cryptography and three-pass protocol make it less vulnerable to attacks and faster in computation. The incorporation of Diffie-Hellman in the protocol allows for secure and efficient key exchange, making it a popular choice in modern cryptography applications.

1. ENCRYPTION SCHEME

DHIES or Diffie-Hellman Integrated Encryption Scheme as it stands is acryptographic which combines the Diffie-Hellman key exchange algoritm with an integrated encryption algorithm. DHIES allows two parties to use a Diffie-Hellman exchange which helps to share key that is then used to secure and decrypt messages. This scheme provides an efficient and secure way of exchanging encrypted messages between parties without the need for a public key system. The DHIES scheme uses a distributed network algorithm that uses a combination algorithm to encrypt the actual message, and a Diffie Hellman key exchange is used to exchange keys for encryption. Encryption algorithm used in DHIES may be a combination encryption algorithm, but the most commonly used algorithms include AES and 3DES.

The use of symmetric encryption allows messages to be encrypted and decrypted quickly and efficiently.The DHIES concept also provides a way to use digital signatures to identify the sender. The sender's private key is used to generate a digital signature, which can be validated using the sender's public key this whole thing comes under Public key infrastructure. This gives the recipient confidence that the communication originated by the sender and was not tampered or altered with during transit.

This provides an additional layer of security by using Diffie-Hellman key exchange in DHIES. Shared keys generated by the Diffie-Hellman are private to the two communicating parties, making it difficult for attackers to intercept and decrypt messages. The DHIES concept also uses a technique called key derivation, which ensures that the symmetric key used for encryption is derived from Diffie-Hellman encryption to be secure and effective.

The DHIES concept now offers a safe and effective way to exchange secrets between parties. Use the Diffie-Hellman switch to provide secure communication and use interoperability to ensure efficient communication. DHIES is used in many applications such as email encryption, file encryption and secure messaging.

1. ZERO KNOWLEDGE PROOF

Zero-knowledge proof (ZKP) is a technique that uses cryptography and allows for parties in an connection to prove to the other party that they have certain information and don’t have to rely on sharing the important information itself. This is especially useful where privacy is important and parties want to avoid sharing sensitive information. Deffie-Hellman key exchange is often used in the ZKP process because it allows both parties to create a shared secret without revealing it to others. This secret teaching can be used as a basis for ZKP.

In ZKP protocol using Deffie-Hellman, one member of the conversation wants to prove their identity to the other member that it knows about the shared key without personally revealing the key. An example of a ZKP protocol using Deffie-Hellman is the Schnorr authentication protocol. In this process, the witness and witness agree on a generator g and the prime number p, and each generates a random value of x. The prover then calculates gx modulus p and sends to prover, which generates a random match value e and returns it to the prover. The witness then calculates the (x + ae) modality (p-1) and sends it to the witness.

The validator then calculates g^(x+ae) mod p, and if that value matches the original value sent by the prover, the prover makes sure that the proofreader knows the key x. The ZKP protocol using Deffie-Hellman has applications in many areas from secure communication to authentication and authorization in computer networks. They are particularly useful when parties want to prove themselves or have secrets without revealing sensitive information. The ZKP protocol can also utilized for creating digital-signatures, electronic cash machines, and even cryptographic systems which can help in authentication and authorization[11].

1. PROPOSED SYSTEM
2. INTRODUCTION

The Diffie-Hellman protocol has already been extensively studied in fields surrounding cryptography since its inception. Research has been focused on various aspects of the technique, including security analysis, operational efficiency, flexibility and continuity, and implementation in various scenarios. The security analysis has been a significant area of research, with researchers identifying various attacks that are common as MITM attacks, micro-attacks, and selected ciphertext attacks, and proposing countermeasures to mitigate them. Methods such as proof of security and data security theory are used to evaluate the security of the system. In addition, studies have been conducted to evaluate the stability of the key-exchange protocol itself in special situations such as malicious individuals, external attacks, or quantum computers.

The effectiveness of the Diffie-Hellman key exchange in todays post quantum computin era is also a subject of ongoing research, with researchers proposing many ideas to improve the performance of the process. These include speeding up algorithms[12], using special properties of finite fields or elliptic curves[13], precomputing or caching, parallelization, and hardware acceleration.

The objective of this project is to create a secure channel to transmit and receive various forms of data, including text messages and images. To accomplish this, the Diffie-Hellman and AES encryption methods will be employed to encrypt the data prior to its transmission to the recipient. The data will be transmitted and stored in the cloud, with encrypted channels such as SSL being applied to the channel and key information before it is sent.

It is widely recognized that security risks are constantly increasing. There is no guarantee that exchanging information with someone else will be conducted securely and without the possibility of being hacked or cracked. Therefore, the proposed project aims to explore encryption techniques that can help conceal the true key and enable safe data transmission.

The proposed solution is a cloud-based chat application that utilizes the Diffie-Hellman key exchange which helps in establish secure communication between users. The app will allow real-time exchange of text messages over the cloud while ensuring data security. The Diffie-Hellman protocol will be used to facilitate key exchange process.

The main objectives of the proposed solution are:

> Develop secure key exchange mechanism using the Diffie-Hellman algorithm for secure communication between users.

> Develop authentication and authorization mechanisms to limit access to legitimate users.

> Create robust management mechanisms for secure storage, distribution, and retrieval of text and files.

> Implement data encryption and integrity mechanisms using symmetric encryption algorithms to ensure secure transmission and storage of messages.

> Design an intuitive user interface that meets security requirements.

> Develop a scalable and robust system architecture capable of accommodating a large number of users while handling network failures or disruptions.

> Ensure compliance with relevant data privacy and security regulations and standards.

The cloud chat application will be developed using both front-end and back-end technologies. Web development frameworks, authentication and encryption libraries, and cloud storage services will all be used. The Diffie-Hellman key exchange protocol algorithm will be used to establish shared secret keys, while symmetric encryption algorithms like AES will be used to encrypt and ensure data integrity. Proper authentication and authorization mechanisms will be employed to verify user identities, and key management mechanisms will be developed to ensure secure storage and handling of shared secret keys. Finally, the application will be designed to be user-friendly, efficient, and scalable, with robust error handling and recovery mechanisms to ensure reliability and availability.

1. CLOUD IMPLEMENTATION

The protection of user's data in cloud storage is crucial, and encryption is often used to achieve this. However, it's important to ask whether the data is actually being encrypted when stored in the cloud, also the cryptographic tool or algorithm used and key length is being used. Some cloud service providers (CSPs) provide encryption, but the quality and security of their chosen algorithms may vary. It's essential to ensure that the encryption algorithm being used is secure and able to support various encryption use cases, such as database and file system encryption.

To address this issue, a solution is proposed that involves using a symmetric cryptosystem with a 128-bit key length, most popular amount then is AES which also stands for Advanced Encryption Standard, in combination with the Diffie Hellman algorithm[14]. In the case of high data quantities, this method provides both speed and efficiency. The key length is sufficient to give adequate security while being computationally practical for today's computing power.

The user would initiate this encryption solution between their cloud-based apps or web-apps and database servers also in cloud. This would include encrypting the data before uploading it to the cloud with the AES method. The data can be accessed by requesting programmes after decryption at the user's end, but it is never written anywhere in the cloud. This technique provides transparency and convenience of integration without needing application modifications.

A physical key management server can be deployed on the to assure secure storage of the encryption keys. This guarantees that the keys stay under the control of the user and are never exposed whether in storage or if transmission is needed. The Diffie Hellman algorithm is used for key exchange for the purpose of authentication.

1. AUTHENTICATION

The Diffie-Hellman key protocol is key agreement protocol, but you can also use it as a basis for authenticated protocols and to provide secrecy as a new key is generated for every session .The function of the Diffie-Hellman Algorithm Authentication Module is as follows:

***New cloud service registration:*** Businesses or consumers that require various cloud services must register before hand by providing a user ID, password, and other information. This information is then used to confirm whether the user is real and who they say they are by sending a brief text message with the key required to create an account from the cloud. A key is generated and transferred to the user's device through a Deffie-Hellman key exchange protocol when they enter their user ID and password. This key is only valid for one session as a token at a time and is erased when the connection is closed.

***Utilizing cloud services:*** To utilize cloud services, users must enter a username and password. Once the passwords and username is authenticated. A new key is produced and transmitted to the user's application wheather he is using it on mobile or web-browser through the Diffie-Hellman key exchange using the secure channel created by the cloud-client architecture. After successful authentication, the user may access the data and all cloud services if the key matches the key produced by the Diffie-Hellman method.

1. CIA TRAID

The CIA Triad paradigm of information security consists of three components: availability, secrecy, and integrity. concerning cloud computing. When sensitive data is transported or kept in the cloud, privacy safeguards should be in place to prevent unauthorised access or disclosure. To maintain anonymity, cloud providers frequently employ security measures like as logging, access limits, and network segmentation.It is critical in the context of cloud computing to guarantee that data is not tampered with or altered during transmission or storage. The correctness and completeness of the information are referred to as its integrity. Cloud service providers will utilise techniques such as checksums, digital signatures, and hash functions to assure data integrity.

1. **Data Confidentiality**

Checking and analyzing the data privacy features of our plan by comparing it to existing encryption methods and sties that use public keys cryptography for data encryption, such as the Advanced Encryption Standard or Triple Data Encryption Standard. Data is encrypted according to the hashing algorithm and the special salt which is generated for every password and cloud service providers are prohibited from accessing the data as they do not have a key known only to the data subject. This component ensures that our proposed system is considerably more secure from a security and privacy standpoint as only the user and can access them and that too is encrypted in transmission where the transmission channel itself is encrypted. The layers of encryption helps maintain to data confidentiality.

1. **Integrity**

Maintaining the correctness, integrity, and consistency of data throughout its lifespan, including creation, storage, modification, and destruction, is referred to as integrity. Data security is critical to preventing unauthorised data modifications, deletions, or additions that might result in inaccurate judgements or the application of relevant data impacts. Any modifications to the data will result in a different value, suggesting that the data has been tampered with. Since the data travels through secure channels which are encrypted from our proposed cloud-client architecture the data integrity is maintained. The data if tampered with on even the cloud end will lead to a flag since the person would not be able to connect as hashes won’t match. Which though is highly unlikely but this will make sure the cloud provider themselves don’t alter the data.

1. **Authentication**

The proposed scheme employs a Two Factor Authentication protocol, which involves the user's password, set during registration, and a key generated using the Diffie-Hellman algorithm, which is then sent to the user's console weather he is on mobile or web-application. When The password and username provided by the user match with those in database after decryption, then the user is granted access to cloud services. Therefore, Authentication in our proposed scheme is rigorous, simple and secure.

Availability is given as the cloud have redundancy on both data and power so all the important pillars or security are present and stand strong in our implementation of text transfer.

1. SECURE TEXT TRASFER

In today's digital age, secure transfer of personal and sensitive information is a critical requirement. Cloud computing is a popular platform for storing and sharing data due to its convenience and accessibility. However, cloud storage and transfer of information come with inherent risks that cannot be ignored. Encryption is a widely used technique to secure data while it is being transferred and stored in the cloud. The Diffie-Hellman key exchange is a very popular algorithm present in one form or other that enables two parties to generate as well as exchange a secret key without any prior set of communications, making it a suitable algorithm for secure communication over a public network.

The proposed project aims to develop a cloud-based chatting application that uses the Diffie-Hellman algorithm for secure authentication and encryption of messages. The application provides a secure channel for communication over the cloud, which helps in maintaining the confidentiality as well as integrity of messages transmitted.

Once the shared secret key is created, transmitted and connection is establish. The text messages, files as well as images can be encrypted and transmitted over the cloud to the receiver in a secure manner. Before sending the any information to the receiver in the cloud, the system encrypts it using the AES hashing algorithm. AES is a symmetric encryption algorithm that uses the very same key key to encrypt as well as decrypt data which was previously shared between the parties.

Proper authentication and authorization mechanisms are implemented in the proposed cloud-based chatting application to ensure that the legitimate users which are authenticated beforehand can access the cloud and participate in secure communication. Key management mechanisms are also developed for secure storage and distribution of text and files.

The proposed application is designed to be user-friendly, efficient, and scalable, with robust error handling and recovery mechanisms to ensure reliability and availability. The system architecture is developed to handle a large number of users and potential network failures or disruptions. The application is compliant with relevant regulations and standards for data privacy and security, making it a reliable and trustworthy platform for secure communication over the cloud.

1. CONCLUSION

In conclusion, the Diffie-Hellman key exchange algorithm in a cloud-based environment provides a secure and efficient method for text transfer. By utilizing symmetric encryption algorithms AES, to secure data and establish shared secret keys, sensitive information can be transmitted and stored securely. Proper authentication and authorization mechanisms further ensure the integrity of the communication between two parties and they can mutually trust each other to share important and confidential information.

Furthermore, a user-friendly interface and robust system architecture can make the application scalable and reliable, capable of handling a large number of users and potential network failures or disruptions. Adherence to relevant data privacy and security regulations and standards further enhances the security of the system.

The proposed solution provides an effective means of securing text transfer in a cloud-based environment, mitigating potential risks and protecting sensitive information.

ACKNOWLEDGMENT

Thank you to everyone who helped make the "Cloud-Based Secure Text Transfer Using Diffie-Hellman Key Exchange" project a success.

First and foremost, I'd want to thank Mr. Namit Chawla, my programme director for his invaluable advice and assistance during the project.

His knowledge and expertise are critical in deciding the project's direction and guaranteeing its effective conclusion upon completion.

I'd also want to thank the participants who offered helpful recommendations and valuable advice throughout the project's development phase. Their feedback assists in identifying possible issues and making essential system adjustments for making a secure method of transferring data over cloud.

Finally, I'd want to thank Chandigarh University for providing me with this chance of creating a minor project as my coursework which helped me gain knowledge and skills. This project’s knowledge and skills I picked up while completing this research paper will surely benefit my future career.

REFERENCES

[1] Prevention of the Man-in-the-Middle Attack on Diffie–Hellman Key Exchange Algorithm, DOI:10.1007/978-981-15-7834-2\_58,Samrat Mitra, Samanwita Das, Malay Kule

[2] Weak keys of the Diffe Hellman key exchange I January 2005 Amit Kalele, Virendra R. Sule

[3] A Diffie-Hellman Key Exchange Protocol with Entities Authentication, Om Pal , Bashir Alam, ISSN:2319-7242 Volume 6 Issue 4 April 2017

[4] The Simplest Protocol for Oblivious Transfer Tung Chou, Claudio Orlandi, Proceedings of the 4th International Conference on Progress in Cryptology -- LATINCRYPT 2015 - Volume 9230August 2015

[5] Delay and cost performance analysis of the diffie-hellman key exchange protocol in opportunistic mobile networks March 2018 DOI:10.1088/1742-6596/978/1/012016, Bambang Soelistijanto, University of Surrey, Vinsen Muliadi

[6] Security Issues in the Diffie-Hellman Key Agreement Protocol January 2002 IEEE Transactions on Information Theory 22, Jean-francois Raymond, Anton Stiglic

[7] ECDH-ECC: A Combination of Elliptic Curve Cryptography and Diffie Hellman Based Cryptography technique for Big Data Security October 2018 DOI:10.13140/RG.2.2.27179.18722, Dr.Subhashini Radhakrishnan, Adilakshmi Vadavalli

[8] DHIES: An encryption scheme based on the Diffie-Hellman Problem January 1998, Michel Abdalla, Mihir Bellare, Phillip Rogaway

[9] Improved Diffie-Hellman Algorithm for Network Security Enhancement, November 2018, DOI:10.13140/RG.2.2.22694.78404 Vishal Garg, Rishu

[10] HMQV: A High-Performance Secure Diffie-Hellman Protocol,Advances in Cryptology – CRYPTO 2005 Conference paper, Hugo Krawczyk

[11] Implementation of Modified Diffie-Hellman Key Exchange Algorithm Using Zero Knowledge Proof, December 2020Solid State Technology 63(6):14055-14070, Mr Chakradhar, Shasidhar K, Mr B Girirajan, Mr Ch

[12] Exponentiation Speed Up For Diffie-Hellman Key Agreement Protocol January 2006, Shay Gueron

[13] Elliptic Curve Cryptography-Diffie-Hellman Technique Extended ForMultiple Two Party Keys At A Time May 2011, Vankamamidi srinivasa naresh, Nistala V.E.S. Murthy, V.V. Narasimha Rao

[14] Use of Digital Signature with Diffie Hellman Key Exchange and AES Encryption Algorithm to Enhance Data Security in Cloud Computing April 2013,DOI:10.1109/CSNT.2013.97,Prashant Rewagad, Yogita Pawar

[15] The Past, Evolving Present, and Future of the Discrete Logarithm November 2014, DOI:10.1007/978-3-319-10683-0\_2,Antoine Joux, Andrew Odlyzko, Cécile Pierrot

Fig 1 Reference: <https://cryptoquantus.com/2019/09/17/ix-diffie-hellman-merkle-key-exchange/>

Fig 2 Reference: <https://www.simplilearn.com/ice9/free_resources_article_thumb/intro_2.PNG>