Hashed chipertext(H) and chipertext(E) downloaded to server

Unbundle E, H

Generate hash H1 of E

If H1==H

Decrypt E

Decryption time

Login

Authentication by SHA-I

Input file, F

Key generation by ECC



ECC key combined with Diffie-Hellman

Hash Encrypted Text, H by SHA-256

Encryption, E with AES

Combine E, H then uploaded on server

Encryption time

Analysis

Storage

**Chapter 5**

**Proposed method**

*Contributions*

The following are the main contributions of this thesis:

* We are using five ways of protection structure in the proposed architecture. First of all, for key exchange step to generate keys, Diffie Hellman and ECC algorithm is used. Then for authentication digital signature is used, there after user’s data file is encrypted or decrypted using hybrid encryption algorithm.
* We propose a hybrid model by combining four algorithms AES, ECC, Diffie-Hellman and SHA-256. In which key generation of AES will be done with the help of ECC. In simple words, we will not be using the key generated by the AES algorithm; instead of AES we will use ECC for generating the key so that its key size is reduced.
* We know that in symmetric/asymmetric encryption, a public key or private key is used for encryption and decryption of data. Thus, this process needs a large key size and requires a lot of computational power. The proposed hybrid algorithm (AES-ECC-DH-SHA) is used to enhance the security of system in less time by solving the problem of key size and it helps to reduce the computational power for memory optimization.
* Also in symmetric key cryptography private key must share between sender and receiver so the key is vulnerable to man-in-the middle attack. That’s why we used ECC for key generation because it is hard to crack by third parties.
* In proposed architecture, instead of hashing the plain file we decided to hash the encrypted file. This is prepared to prevent approach to the plain file until the consistency of the file is checked.
* We also present an algorithm for our proposed framework which describes how the public key is generated using ECC algorithm, how Diffie-Hellman will help in generating a shared secret which is then combined with ECC key and how encryption/decryption is done using AES and SHA-256.
* At last we will analysis of proposed model is done on the basis of different metrics like storage, encryption time and decryption time, correlation and avalanche effect.

**A. Execution Steps:**

1. Sign up

2. Login from TCP

2.1 key Generation – Eliptic Curve Cryptography

2.2 Key Exchange – Diffie Hellman

2.3 Digital Signature –SHA-I

3. Uploading / Downloading Data Encryption- Hybrid Approach

4. Data is stored / retrieved from Storage server

5. Logout.

# Proposed Framework

In this section, we will be providing the design details of the proposed approach. We emphasize the significance of combining both symmetric and asymmetric cryptographic techniques like ECC, AES, Diffie-Hellman, SHA-256 and concentrate on the algorithms that will be used in our proposed hybrid approach.

# In the proposed architecture, we are going to use five ways of protection scheme. To begin with, to generate keys for key exchange step, ECC and Diffie Hellman asymmetric algorithm are employed. Then for authentication digital signature is used, subsequently using hybrid encryption algorithm for user’s data file is encrypted or decrypted. On cloud server by double encryption with hybrid algorithm data will be uploaded. At the begin data will be encrypted using Advance Encryption Standard algorithm (C) and once more hashing will be apply on that encrypted file by SHA-256 (H) at last encrypted file and hashed file (C, H) will be uploaded on cloud server and in the same way from the cloud server data will be downloaded by decrypting the file as exactly opposite of encryption process. During decryption process at the first unbundles the payload (H, C) then apply hashing on encrypted file H(C) there after integrity check (is H==H(C)) if the value of hashed encrypted file is equal to hash file then only decryption process is done. At the server end to provide trusted network all this is implemented. For storing user data file two distinct servers are maintained for the same reason, one for encryption process known as (trusted) computing platform and another known as storage server where data file uploaded. When a consumer wants to upload a data file to the cloud server, first at the time of login using Diffie Hellman key exchange key are exchanged, subsequently using digital signature the client is authenticated. Finally using hybrid encryption algorithm user’s data file is encrypted and only then it is uploaded to Cloud Storage server. From Cloud server, the client can download the same file. When a user logins, initially encryption keys are exchanged, file to be downloaded is selected, using digital signature authentication takes place and integrity check between hash file and encrypted hash file at last AES algorithm is used to decrypt the saved file and finally client is accepted to access the data file.

# Research Methodology

In this chapter, we provide design of the research that has been followed by the thesis. The research design for the proposed method is also discussed in the next section. Figure [4](#_bookmark9) gives an example of the conventional research methodology followed in wide-ranging, from reviewing present schemes to proving the proposed one.

From the figure [4](#_bookmark9) it can be seen that the flow of this thesis follows the conventional research methodology. Initially, we started by investigating already existing schemes in the literature. After investigating, in these existing schemes we recognized several limitations. A key limitation of this research is that, we discovered that computation overhead cost for existing schemes is larger as is the time required to compute them. Several other limitations were found; however, We not solve all the limitations but we try to solve only a few of the main limitations by proposing our new hybrid approach (Symmetric key cryptographic technique for data encryption, Asymmetric key cryptography techniques for key management and Hashing technique for integrity maintain ), we solve which enhance the security of the data over the cloud storage. For our proposed hybrid scheme Experimental setup is also created and comparison can be done between planned scheme with other methods and hybrid scheme as well. We obtained that our proposed hybrid scheme do better than other security schemes in terms of efficiency and performance.

Identifying limitations of existing schemes

Proposed new hybrid approach

Experimental setup

Findings and conclusions

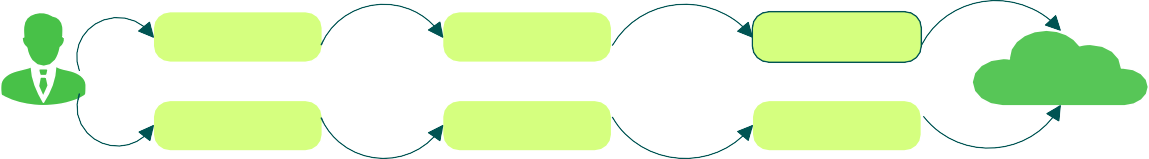
Review of existing security techniques

**Figure 4.** Methodology used in the paper

From the figure [4](#_bookmark9) it can be seen that the flow of this thesis follows the conventional research methodology. Initially, we started by investigating already existing schemes in the literature. After investigating, in these existing schemes we recognized several limitations. A key limitation of this research is that, we discovered that computation overhead cost for existing schemes is larger as is the time required to compute them. Several other limitations were found; however, We not solve all the limitations but we try to solve only a few of the main limitations by proposing our new hybrid approach (Symmetric key cryptographic technique for data encryption, Asymmetric key cryptography techniques for key management and Hashing technique for authentication and integrity maintain), we solve which enhance the security of the data over the cloud storage. For our proposed hybrid scheme Experimental setup is also created and comparison can be done between planned scheme with other methods and hybrid scheme as well. We obtained that our proposed hybrid scheme do better than other security schemes in terms of efficiency and performance.

***Defining ECC and AES***

ECC safeguards the data from unauthorized access and cryptographic technique that uses following asymmetric key encryption .Also, ECC is a well-known cryptographic technique. The security of the ECC ensured by uses pairs of public and private keys. ECC uses two dimensional fields as prime and binary fields. If we are using this cryptography technique so hacking is not straightforward because it establishes a relation among primary and binary fields and this relation in ECC cannot be understand by unauthorized people and it also uses the improved operations. The most important aspect of ECC is small key size. On data for to discover the appropriate field for the implementation of cryptography the safety measures maximum number of points can assists. Selects the first number of the field by performing the first operation and then based on the data yields the large number which is between 0 to Z. for The generation of the key, specifically, ECC is employed and the complexity of the procedures is decreased. The enhancement of ECC is considerably more than other cryptographic techniques due to its low-key size. This paper uses the ECC techniques to space enhancement and optimize memory [[28](#_bookmark53)].



Encryption

Verification

Cloud Server Provider

User

Request for upload

Cloud Server

Decryption

Verification

Cloud Server Provider

Request for download

Securing data in Cloud Computing Using Elliptic Curve Cryptography

To safeguard the data for the encryption and decryption process AES uses solitary one key. AES is one of the types of cipher text which employs the block cipher. On cloud storage like searching on cloud storage, statistical analysis, and others like these it has many performance procedures which are restrictive on cloud storage. To enhance the security instructions over cloud storage it is the extensively used strategic algorithm over cloud computing. it is easily implemented as well as being time compatible with the cloud storage accessibility of data that’s why this paper uses the AES algorithm [[29](#_bookmark54),[30](#_bookmark55)].

**Hybrid Approach of ECC, AES-256, Diffie-Hellman and SHA-256**



Input File

Encryption with AES

+

Encrypted Text Uploaded to Server

Encrypted Text Downloaded to Server

+

Key Generation by ECC

Analysis

Decrypted Text

Storage

Encryption/ Decryption on Time

The cipher text will be generated Once the key size is set the encryption and decryption of data. The combined effect of all four AES,ECC,SHA and Diffie-Hellman is suitable for the proposed technique at cloud storage to get the secured system and secure key transection . This will help us to reduce the storage size with secure data and also validates data integrity.

Encrypted Text

ECC key combined with DH

**Figure 5.** Representation of Hybrid Approach.

In the above figure, it can be clearly seen that AES along with ECC and Diffie-Hellman effectively secures data over cloud storage. The novelty of the proposed method can be clearly seen in the new proposed diagram, in which there is secure transmission of user data to server and then storage mechanism is even secured due to encrypted data. Moreover, novelty can be determined in terms of computational cost and time. However, attack prevention can be done in the following way: for example, if an attacker wants to attack on the user side in order to gain user personal information or for some other purpose, in the proposed approach, once the user uploads the input file, the file is converted into encrypted text.

In our proposed architecture, we made employment of the secure hybrid cryptographic technique (SHCT), a technique that takes benefit of the strong point by combining of two or more encryption systems into a single platform. To employment the strength of one encryption scheme to address the weakness of another type is the justification behind the usage of SHCT. Our hybrid cryptosystem comprises of SHA-I,AES 256, ECC, Diffie-Hellman, and SHA-256. In our system, instead of hashing the plain file we determined to hash the encrypted file. This is prepared to avoid access to the plain file until the integrity of the file is confirmed. The cryptosystem comprises of two architectures: receiver’s architecture and sender’s architecture.

With the help of AES along with SHA-256 encryption, so the text is fully encrypted and hashing technique applied on that encrypted file. Consequently, if an attacker performs an attack and by some means obtains the user-uploaded file, then it is worthless because the information was already encrypted and hashed while uploading the file. In the same way, on the other end, if an attack is operated, the attacker is not able to decrypt the encrypted file and hence it uses SHA-256 for integrity maintain so the data are secured from attacks.

* 1. *Algorithm for the Proposed Framework*

*1.1.1.*Digital signatureSHA-I for authentication

|  |
| --- |
| *Step 1:-* Padding of Bits is first step of SHA-I, at the last part of the legitimate message add Padding bits and the length is multiple of 512 and 64 bits.  *Step2:-* The second step is Appending length in this step the non-including length is computed.  *Step3:-* Separate the Input data file into 512-bit blocks  *Step4:-*Initialize chaining variables is fourth step. In this stage we initializing chaining variables here we initialize 5 chaining variables of 32 bit each=160 bit of total.  *Step5:-*Process Blocks i) Copy the chaining variables  ii) Divide the 512 into 16 sub blocks  iii) Process 4 rounds of 20 steps each |

1.1.1. By Elliptic Curve Cryptography generation of a Public Key

**Box 1**

|  |
| --- |
| *Step I.* Choose one number n in the role of the prime number.  *Step II.* Choose one number for the generation of the public key such as n (a) Where n (a) < n  *Step III.* Calculate the point on the curve such as G Where G > n  *Step IV.* Computation of public key is: P = n (a) ∗ G  *Step V.* Reappearance the public key P after computation. |

1.1.3.Reliable key exchange by using Diffie-hellman Algorithm

**Box 3**

|  |
| --- |
| *Step I*. Sender and Receiver approve to an agreement on a prime number p, q considering that it’s primitive root.  *Step II*. Sender and Receiver select their so called private key ‘a’ and ‘b’ which is only comprehended to themselves correspondingly.  *Step III.* Public key of the Sender is A= qa mod p.  *Step IV.*  Public key of the Receiver is B= qb mod p.  *Step V.* Thepublic key is exchanged by Sender and Receiver. Now Sender has B(Receiver’s public key) and Receiver has A(Sender’s public key).  *Step VI.* Sender computes Ba mod p= qba mod = S.  *Step VII.* The receiver computes Ab mod p= qba mod p= S.  Consequently, the sender and receiver obtain ‘S’ as their shared secret key. |

1.1.2. Encryption and Decryption by Using proposed architecture

**Box 2**

|  |
| --- |
| *Step I.* Take the input file  *Step II.* Now add the key generated by ECC which is the public key.  *Step III.* AES encryption is performed on the input file by using the public key which is generated by ECC.  *Step IV.* The encrypted file is uploaded on the server after the encryption by AES.  *Step V.* Once the file is uploaded then it will be downloaded at the server, and then file is translated by using the public key given by ECC so that the original file is decrypted.  *Step VI.* The performance of the system depends on the combined effect of ECC and AES, such as the storage space optimization and enhancement of the security services over the cloud server. |

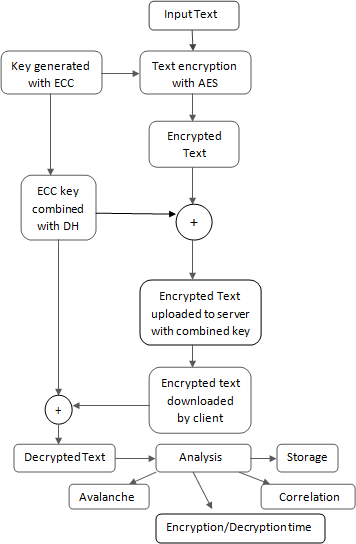
1.1.3. Encrypt and decrypt file by AES

|  |
| --- |
| ***ENCRYPTION TIME:***  *Step I.* By using a substitution table The first step of an AES encryption cipher is data substitution. The input state array is XORed with the first four words of the key schedule before any round-based processing for encryption can start.  *Step II.* The second step is to change the data rows and the third step is to mix the columns.  *Step III.* Each round consists of the following four steps for encryption: Substitute bytes, Shift rows, Mix columns and Add a round key.  *Step IV.* Lastly, we XOR the four words from the key schedule with the output of the previous three steps. The last step is done using a different part of the encryption key on each column and in the last step in encryption the mix columns step does not have.  ***DECRYPTION TIME:***  *Step I:* The similar thing occurs, during decryption, apart from that with the last four key schedule words we now XOR the ciphertext state array.  *Step II:* for decryption the following four steps consists of each round: Inverse shift rows, Inverse substitute bytes, Add round key and Inverse mix columns.  *Step III:* At last, from the key schedule we XOR the four words with the output of the preceding two steps. The last step in decryption does not have the inverse mix columns step. |

1.1.4. Apply hashing algorithm by SHA-256 Algorithm

|  |
| --- |
|  |

**How Hybrid Algorithm work with text file-**



### Figure 3. AES-ECDH Encryption/Decryption

**Step 1:** Different sizes of diverse data files are taken as input, Size of files in Kb. Group of files accepted as input are license.txt, new.txt, new1.txt, readme.txt and example.txt. From the server client will demand a data file and afterwards for Encryption the request of the client data file will be selected by server.

**Step 2:** As soon as receiving data file as input, Elliptic curve will produce different pairs of public key and private key. Encryption is employed by Advance encryption standard and by using one of the key pairs AES will encrypt the data file generated by Elliptic curve defined. Starting from the distinct key sets with the server say **‘d’** one key will be retained secret and to the client say **‘e’** one key will be secret.

**Step 3:** Between both of the communicating parties by yielding a successful key agreement the shared secret between client and server will establish by Elliptic Curve Diffie – Hellman. Let’s investigate among client and server in order to generate the shared secret how the agreement will be composed.

Therefore, if a third party would like to breakdown shared secret, they have to crack the discrete logarithm problem and just for a specific session the shared secret will be in between client and server. When encrypting of the input data file will be done by AES on the other hand between client and server the key agreement will be performed by Diffie- Hellman. If successfully key agreement will be done only then client will be capable to decrypt the encrypted data file.

**Step 4:** By Elliptic Curve Cryptography the key will generated and using by AES to encrypt the message. Once encryption is completed, then the encrypted text is uploaded on the cloud server along with the combined key *i.e*. from Diffie-Hellman another key as shared secret gained. From the server by using the combined key formed by DH and ECC consumer will download that encrypted file and decrypt that data file except merely after the successful key agreement among client and server that will establish a shared secret.

**Step 5:** Client will have the original data file after the decryption is successfully completed. Eventually after encryption and decryption is successfully done, on the basis of specified metrics analysis of SHCT is performed .Certain metrics like storage for encrypted files, encryption time, decryption time, avalanche effect and correlation. The time taken to convert the original text file into the cipher file is called Encryption time; Decryption time represents the time taken to convert the cipher file reverse to the original data file. The size of cipher file or encrypted formed after encryption signifies that storage is essential to measure it if coming toward cloud. In the cipher furthermost importantly Avalanche effect by making a bit modification in the original input file that will instruct about variation file and the reliance between and the original file and the cipher file will signified correlation *i.e*. whether the correlation among the files is linear but decreasing or linear and increasing, correspondingly if correlation is less then both original file and cipher file are contradictory and it is incomprehensible for intruder to distinguish original text from cipher text.)

**3.1 Sender’s Architecture (Encryption)**

As presented in Figure 5.1 is described the flow of program of this sender’s architecture is as follows:

1. A. Generate shared secret key by using ECC.

B. By using Diffie-Hellman establishes secure and reliable connection between two parties and share shared secret key.

C. Authentication done with digital signature using SHA-I.

D. Take any text file (data file) by means of input.

2. With the help of shared secret key by using AES 256 algorithm generate a ciphertext by encrypting the chosen file.

3. By applying SHA-256 Hash the ciphertext.

4. Combine the encrypted file and the hashed ciphertext as a payload and send this payload to the receiver.

**3.2 Receiver’s Architecture (Decryption)**

Figure 2 demonstrates the flow of the process of Receiver’s architecture. Converts the encrypted file into its original form is the decryption phase, for the convenient to the receiver so the receiver can approach or read the file contents. It is described as follows:

1. The sender sends payload and in this phase unbundled the payload that received from the sender. The payload consists of two parts:

a. The first part is the hashed ciphertext (H) created by the sender.

b. The other part is the ciphertext (encrypted file E).

2. By applying SHA-256 Hash the ciphertext (H1).

3. Evaluate the sender hashed ciphertext (H) with the receiver hashed ciphertext (H1):

a. If the evaluation obtains both hashes (H and H1) to be accurate then: jump to step 4.

b. Otherwise: terminate the procedure and discard the records.

4. Generate shared secret key by using ECC.

5. By using Diffie-Hellman establishes secure and reliable connection between two parties and share shared secret key.

6. To obtain the plain file decrypt the ciphertext, by using the AES 256 algorithm and shared secret key.

Select File (F)

Generate Shared Secret key (K) by ECC

Generate Chipertext (C) By AES C= Encrypt (F,K)

Payload on server (H,C)

Generate Hash (H) By SHA H=Hash-function (C)

Authentication by SHA-I

Hash Chiper (H)

Chiper (C)

Generate Hash (H1) H1 = Hash-function(C)

Generate Shared Secret Key (K)

Restore File (F) F= Decrypt (C,K)

IsH1 = = H?

Unbundle Payload (H,C)

Authentication by SHA-I

yes