Ibn Al-Haitham Journal for Pure and Applied science https://doi.org/10.30526/2021.IHICPAS.2655

Improved Certificate-Based Encryption Scheme in the Big Data: Combining AES and (ECDSA – ECDH)

Omar Salah F. Shareef
omar.alshareef@uofallujah.edu.iq
University of Fallujah, Fallujah, Iraq

Ali Makki Sagheer
prof.ali@alqalam.edu.iq
Qalam University College, Kirkuk, Iraq

Abstract

Big data usually running in large-scale and centralized key management systems. However, the centralized key management systems are increasing the problems such as single point of failure, exchanging a secret key over insecure channels, third-party query, and key escrow problem. To avoid these problems, we propose an improved certificate-based encryption scheme that ensures data confidentiality by combining symmetric and asymmetric cryptography schemes. The combination can be implemented by using the Advanced Encryption Standard (AES) and Elliptic Curve Diffie-Hellman (ECDH). The proposed scheme is an enhanced version of the Certificate-Based Encryption (CBE) scheme and preserves all its advantages. However, the key generation process in our scheme has been done without any intervention from the certificate issuer and avoiding the risk of compromised CA. The Elliptic Curve Digital Signature Algorithm (ECDSA) has been used with the ECDH to handle the authentication of the key exchange. The proposed scheme is demonstrated on a big dataset of social networks. The scheme is analyzed based on security criteria that have been compared with the previous schemes to evaluate its performance.

Keywords: Big Data Security; Certificate-Based Encryption; ECDSA; ECDH; AES.

1.Introduction

The term ‘Big Data’ has been become an essential part of people's lives. The appearance of big data has brought new challenges regarding data security. The collecting, storage, manipulation and retention of huge quantities of data have led to critical security and privacy considerations (Cheng et al.,2017). Accessing the data needs to be controlled to affirm that the non-eligible entities cannot tamper with or access the data. Improving the security and authentication of sensitive data can give the companies new businesses opportunities. Therefore, close attention has been drawn to secure data firmly from any unauthorized access. The main challenge in big data that are using a complicated distributed system, is the multifaceted nature of overseeing wide usage. Verification ought to be overseen by an adaptable, hearty and versatile framework that denies a malevolent client from access big data. Accordingly, new ways to deal with security are required to defeat the security blemishes in the current usage. However, there are two methods to design a secure system and protecting the privacy of the receivers in a communication system; symmetric and asymmetric keys.

The secure cryptosystem is based on effective key management. However, the security of any cryptosystem is dependent upon how securely its keys are managed?

Public Key Infrastructure (PKI) generates and handles a pair of keys in addition to the certificate. The main challenge of managing the digital certificate in public key infrastructure is when we have many nodes in a distributed environment.

To cope with this issue, Shamir (A Shamir, 1984) produced the concept of Identity-Based Encryption (IBE) where the identity was used as a public key, and a third party generates a secret key for the user. This approach eliminates the demand for digital certificates. However, IBE suffers from a key escrow problem, the third party holds the private keys of clients, as a result, it can peek at all communication data of users and successfully disguise himself as any user to sign a message. Alternatively, Certificate-Based Encryption (CBE) overcomes the drawbacks of PKI and IBE.

In 2003, Gentry (Gentry, 2003) presented the idea of CBE. This scheme merges Public-Key Encryption (PKE) and IBE while safeguarding the highlights of schemes. As in conventional PKE, every client produces a pair of keys then asks for a certificate from the Certificate Authority (CA). The significant matter in CBE is a certificate works as a partial private key as well as its work as a certificate. This function produces an effective implicit certificate, thus a recipient wants the certificate together with his/her private key to decrypt a ciphertext, whilst the senders do not need to be concerned about the certificate revocation problem. However, the CBE copes with the limitations of IBE and PKE. The CBE can ignore third-party queries. Then, disregarding a secure channel among the CA and clients. Furthermore, it is avoiding the key escrow problem (since CA does not know the private keys of users). Finally, it fixes the certificate revocation problem.

In recent years, CBE has attracted considerable interest in the community of research and several schemes in CBE have been proposed (Liu and Zhou, 2008; Lu et al., 2008; Galindo et al., 2008; Hyla et al., 2015; Lu and Li, 2014; Le, Kim and Hwang, 2016; Hwang and Le, 2018).

In this paper, we propose an efficient certificate-based encryption scheme that ensures data confidentiality by combining symmetric and asymmetric cryptography schemes. Particularly, using the Advanced Encryption Standard (AES) and Elliptic Curve Diffie-Hellman (ECDH). Moreover, the Elliptic Curve Digital Signature Algorithm (ECDSA) has been used with ECDH to handle the weakness of authentication between the nodes whilst using the ECDH approach. When the system is on a large scale, the task will be computationally intensive. Moreover, the proposed scheme can be supported distributive processing in large-scale key management within a distributed environment.

1. PRELIMINARIES

2.1 ECDSA

ECDSA is the elliptic curve analog of the Digital Signature Algorithm DSA (Johnson et al., 2001). Vanstone proposed the ECDSA scheme as an answer to the request of the National Institute of Standards and Technology (NIST) for feedback on their proposal about the Digital Signature Scheme (DSS) (S. Vanstone, 1992).

DSS is similar to handwritten signatures. The digital signature represents a number based on the private key known only by the signer and on the contents of the signed message, as well as, the signatures should be verified without gain access to the secret key of the signer (Langford, 1995). ECDSA algorithm is defined as follows:

Setup

1. Alice and Bob select a finite field Fq and an elliptic curve E over Fq (E(Fq)).

2. They choose a random base point B ∈ E with order n, such that B generates a large subgroup of E, preferably of the same order as that of E itself.

Key generation

1. choose a secret random integer d in the interval $[2, N]$
2. Computes $Q = dB$
3. Make Q public and keep d secret

Signature generation
Alice sends the signed message to Bob as follows

1. Select random integer k in the interval $[2, N]$
2. Compute $(x, y) = kB$
3. Compute $r = x \bmod n$
4. Compute $e = H(M)$
5. Compute $s = k^{-1}(e + d\,r) \bmod n$
6. The signature for M is (r, s)

Signature verification
Bob verifies Alice's signature (r, s) on message M as follows

1. Compute $e = H(M)$
2. Compute $w = s^{-1} \bmod n$
3. Compute $u_1 = e\,w \bmod n$
4. Compute $u_2 = r\,w \bmod n$
5. Compute $(x, y) = u_1 B + u_2 Q$. If $(x, y) = O$ then reject the signature
6. Otherwise, Compute $v = x \bmod n$
7. Accept the signature if and only if $v = r$

2.2 ECDH

ECDH is a key agreement algorithm that lets two entities generate a shared secret key based on ECC (Anoop, 2001). The following example illustrates how the key establishment will be made.

Suppose A wants to generate a shared key with B through a channel that may be eavesdropped on by a third party. Initially, each entity should agree upon Elliptic Curve domain parameters (q, a, b, G, n, h). Each has a pair of keys; d is a secret key (which is a random integer $< n$, where n is the order of the curve, a domain parameter of the elliptic curve) and a public key $Q = d * G$ (where G is the generator factor, a domain parameter

of the elliptic curve). Suppose $d_A \cdot Q_A$ are the secret/public key of A, and $d_B \cdot Q_B$ are the secret/public key of B.

1. A computes E = $(X_E, Y_E) = d_A \cdot Q_B$
2. B computes F = $(X_F, Y_F) = d_B \cdot Q_A$
3. Since $d_A \cdot Q_B = d_A \cdot d_B \cdot G = d_B \cdot d_A \cdot G = D_B \cdot Q_A$
4. Thus E = F and subsequently $X_E = X_F$
5. As a result, the shared secret is X_E

Because of difficulties regarding discovering the secret key d_A or d_B from the public key E or F, it's difficult to get the shared secret for an outsider.

The algorithm of ECDH Key Exchange is described as following:
Goal: generate a secure shared key
Input: elliptic curve parameter domain
Output: secure shared key E
Step 1:

1. Client A chooses secret random number $X < n$
2. Client B chooses the secret random number $Y < n$

Step 2:

1. Client A computes $PU_A = X * G$
2. Client B computes $PU_B = Y * G$

The two parties share their public keys and the common base point G
Step 3:

1. Client A compute $E = X * PU_B$
2. Client B compute $E = Y * PU_A$

Step 4: Return (E)

3. The Proposed Scheme

In this research, an improved certificate-based encryption scheme has been proposed, and that ensures data confidentiality by combining symmetric and asymmetric cryptography schemes. Particularly, using the Advanced Encryption Standard (AES) and Elliptic Curve Diffie-Hellman (ECDH). Moreover, the Elliptic Curve Digital Signature Algorithm (ECDSA) was used with ECDH to handle the weakness of authentication between nodes while using the ECDH approach. When the system is on a large scale, the task will be computationally intensive.
The proposed scheme will be reduced the workload of the CA by delegating the task of CA to the user level. In other words, the key generation process is done without any intervention from the certificate provider. The keys are generated using the ECDH algorithm and it's derived from the parameters that generate the keys of the certificate and hence avoiding the risk of compromised CA. Accordingly, it can also resolve the problem of violation of any privacy of customers because it resolves the key escrow problem.

Moreover, the proposed scheme can be supported distributive processing in large-scale key management within a distributed environment. The overview of system layers is shown in more detail in **Figure 1**, which presents the block diagram of system operations using the improved CBE.

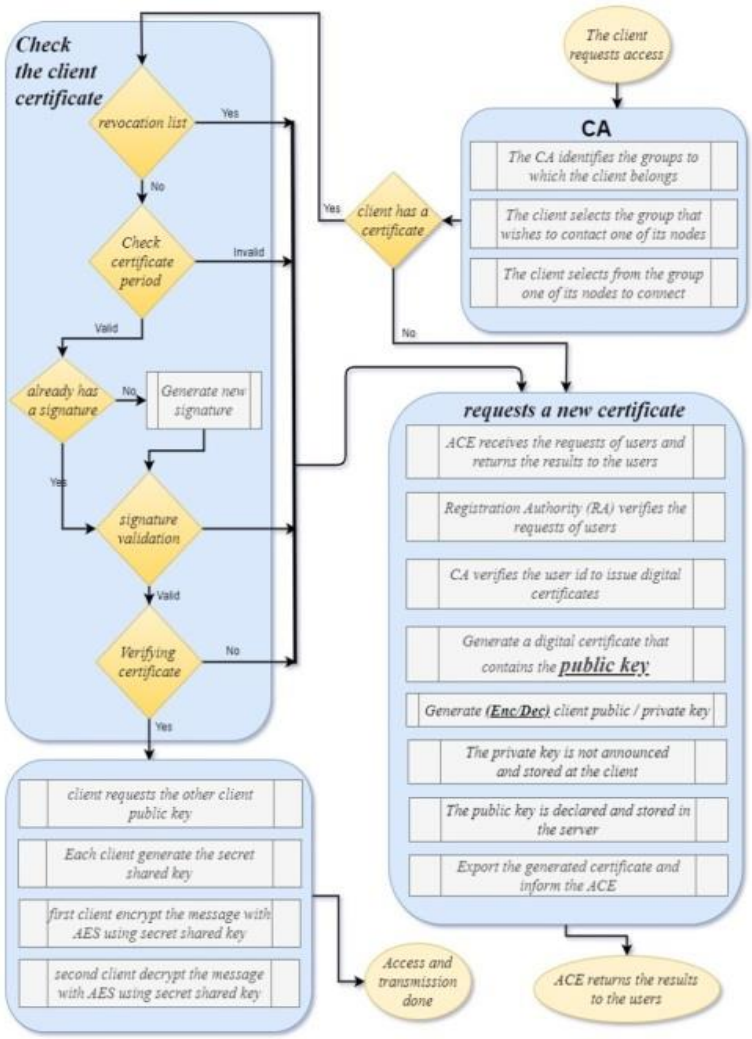

Figure 1 .The Block Diagram of the Proposed Model

By utilizing the benefit of ECDSA, our proposed scheme was built by applying the ECDSA with ECDH that generating the encryption keys. In particular, we designed the setup and certification algorithm by utilizing the key generation as well as the signature generation algorithms of ECDSA. Additionally, by using the key generation algorithm of ECDH, we design the *SetKeyPair* algorithm. Symmetric algorithm AES is used for

constructing encryption and decryption algorithms, in this research AES-128 is used. ECDSA and ECDH are based on some parameters for all network participants.
which:
q = Order of the prime field Fq
E = an elliptic curve y2 = x3 + ax + b defined over the prime field Fq
G = A random non-zero base point in E(Fq)
n = the order of G, typically a prime
h = the cofactor $\frac{|E(fq)|}{n}$
The improved CBE scheme consists of six algorithms:

1. Setup:
ECDSA and ECDH are based on some parameters (params) common for all network participants. Setup run by the CA. It takes the security parameter as input and returns the CA's master secret key *msk*.

2. Certificate *SetKeyPair*:
Input: ID, Domain Parameters.
Output: private key d and secret key Q and master secret key E for encryption/decryption.
private key: $d \in R[1, n-1]$
public key: $Q = dG \in E(Fq)$

3. Encryption/Decryption *SetKeyPair*:
This process has been performed by users without the intervention of the certificate provider. The keys that have been generated will be using the same parameters that are used to generate certificate keys. The shared key is generated using ECDH.

4. Certification:
ECDSA takes ID, params, *msk*, Q and additional identifying information, such as name, as input, and the output is the certificate *Cert* to the user.

A. ECDSA Signature Generation
Input: Domain Parameters, the secret key of the signer d and message M
Output: ECDSA signature (r, s)
k = a haphazardly picked component in $[1, n-1]$ (the session key)
$R = kG$
$r = x(R)$ (the x-coordinate of R) reduced module n
$s = k = 1(H(M) + dr) mod\ n$, H is a cryptographic hash function

B. ECDSA Signature Verification
Input: Domain Parameters, signature (r, s) message M and the public key of signer Q.
Output: accepted or rejected.
$w = s - 1\ mod\ n$
$u = H\ (M)w\ mod\ n$
$v = rw\ mod\ n$
$R = uG + vQ\ \in\ E(Fq\ )$
Accept the signature if and only if $x(R) = r\ mod\ n$

5. Encryption:
Input: ID, params, E, and plain M.

Output: cipher C . AES generate series of AES round keys from E and return (AES_round_key).
Initial Step:
$N_b \leftarrow 4$ (data blocks are of 128 bits)
$N_r \leftarrow$ number of rounds in the cipher (N_r =10 for AES-128)
Step 1:
State ←plain
AddRoundKey (state, w (0, Nb - 1))
Step 2:
For all round do where {1 ≤ round < Nr}
SubBytes (state)
ShiftRows (state)
MixColumns (state)
AddRoundKey (state, w (round * Nb, (round + 1) * Nb - 1))
End for
Step 3:
SubBytes (state)
ShiftRows (state)
AddRoundKey (state, w [Nr * Nb, (Nr+1) * Nb - 1])
cipher_text ← state
Step 4:
Return (cipher)

6. Decryption:
Input: ID, params, Cert, E, and cipher C
Output: plain M
Initial Step:
Nb ← 4 (data blocks are of 128 bits)
Nr← number of rounds in the cipher (Nr = 10 for AES-128)
Step 1:
State ←cipher
AddRoundKey (state, w (Nr * Nb, (Nr + 1) * Nb - 1))
Step 2:
For all-round do where {Nr > round ≥ 1} decrement round by 1
InvShiftRows (state)
InvSubBytes (state)
AddRoundKey (state, w (round * Nb, (round + 1) * Nb - 1))
InvMixColumns (state)
End for
Step 3:
InvShiftRows (state)
InvSubBytes (state)
AddRoundKey (state, w (0, Nb - 1))
Plain← state
Step 4:
Return (Plain)

However, the algorithm of the proposed model is described in detail in **Table 1**.

Table 1: The Algorithm of Proposed Model

Goal: Verify or Generate new Certificate

Input: Access request

Output: The client has access

Step1 *The client requests access.*
Step2 *The CA identifies the groups to which the client belongs.*
Step3 *The client selects the group that wishes to contact one of its nodes.*
Step4 *The client selects from the group one of its nodes to connect.*
Step5 *Check the client certificate*

- *Parse the client certificate*
  - *If the client has a certificate, then go to the next condition*
  - *Otherwise, the client doesn't have a certificate and go to Step 11 or Cancelling request*
- *Check the period of the client certificate*
  - *If valid, then go to the next condition*
  - *Otherwise, expire and go to Step 11 or Cancelling request*
- *Check the certificate signature*
  - *If the client already has a signature from a previous verification, then go to the next condition*
  - *Otherwise, check the signature validation*
    - *If valid, then go to the next condition*
    - *Otherwise, inauthentic and go to Step 11 or Cancelling request*
- *Check the revocation list*
  - *If the client is not revoked in the group to which he belongs, then go to the next condition*
  - *Otherwise, outmode and go to Step 11 or Cancelling request*
- *Check the certificate path until the root certificate*
  - *If the root certificate is valid, then the client has a connection and* ***Done***
  - *Otherwise, invalidate the certificate and go to Step 11 or Cancelling request*

Step6 *The client requests the other client's public key*
Step7 *Each client generates the secret shared key*
Step8 *The first client encrypts the message with AES and using the generated key as encryption key.*
Step9 *The second client decrypt the received message with AES and using the generated key as decryption key.*
Step10 *Access and transmission done.*
Step11 *The client requests a new certificate.*
Step12 *The ACE receives the requests of clients and returns the results to the clients.*
Step13 *The Registration Authority (RA) verifies the requests of clients for a digital certificate and inform the CA to issue it.*
Step14 *The CA verifies the client's id to issue the digital certificates.*
Step15 *Generating a digital certificate which contains its public key.*
Step16 *Generating client's public / private key.*
Step17 *Export the certificate to the server engine.*
Step18 *Store the generated certificate at the certificate service database and inform the ACE.*
Step19 *The ACE returns the results to the clients.*
Step20 *End.*

4. Cryptographic Tool and Dataset

The proposed model was coded in the C# programming language. Also, the open code library "Bouncy Castle" has been used to compose the code. This library is created by the Legion of Bouncy Castle and is a C# usage of cryptographic algorithms.

Friendster social network and ground-truth communities have been used as a dataset for our model (Yang and Leskovec, 2015). Friendster is an online gaming network where clients can create friendship edges with each other. The dataset includes a lot of groups of nodes, each node is either linked to at least one group or is linked to other nodes that belong to different groups. The dataset contains 65,608,366 nodes and 1,806,067,135 edges.

5.System Analysis and Performance

Security Proof

There are two attacks against digital signatures regarding the ECDSA algorithm: a key attack wherein the adversary has a knowledge of the public key and a message attack wherein the adversary has got right of entry to signatures earlier than cracking the function. However, there are various understandings of break a digital signature: recovering the private key, producing an alternative algorithm of signature with a corresponding private key, and then forge a signature for a specific message (Goldwasser et al., 1988).

There are fundamental conditions for ECDSA to make it hard to break and have powerful security (Vaudenay, 2003):

- The discrete logarithm inside the subgroup spanned with the aid of G is difficult. This makes sure it is difficult to resolve the discrete logarithm problem and consequently not possible access the private key.

- The used hash function is a one-way collision-resistant. When could not specify m from $(m) = y$, we can say one-way collision-resistant. However, collision-resistant function has a little opportunity of mapping different messages to be equivalent $(i.e. H(m1) = H(m2))$.

- The generator for k is could not predict. Else, the private key can be acquired using $k, r,$ and s.

Concerning ECDSA, there are two major attacks that either versus the hash function used within the signature generation or against ECDLP (Johnson, Menezes, and Vanstone, 2004). If the algorithm no longer comprises the second bullet point from above, the attacker can discover a collision in the hash function by two messages, and sign a message but claim his signature on the other. ECDLP is described as solving for d in $Q = dG$ within the key generation algorithm. There are common attacks versus ECDLP involving the “Pohlig-Hellman”, “exhaustive seek” and “baby-Step giant-Step” algorithms. Besides these attacks, Pollard’s Rho algorithm has a running time of $(\sqrt{n\pi}) / 2$, in which n is the order of point G. However, this algorithm parallelized and run on r various processors, hence the new running time is $(\sqrt{n\pi})/2r$.

6.Encryption Time Consummation

In this section, a comparative test of the time consumption of our proposed CBE scheme against the RSA algorithm is discussed. Key generation and encryption/decryption of 40 bits' string of characters is tested based on the PC of Core i7 with Frequency 2.2GHz, RAM 8GB, and the operating system is Windows10 with 64bit. These comparisons show the time consumption of the key generation when using ECC of the proposed CBE scheme against RSA as shown in **Table 2**, and the encryption/decryption when using AES of the proposed CBE scheme against RSA algorithm as shown in **Table 3**.

Table 2: Time Consumption of Key Generation (ms)

Key length	Key pair generation	
	RSA	ECC
1024 / 160	237	218
2048 / 224	1440	232
3072 / 256	3377	234
7680 / 384	108718	278
15360 / 512	925284	375

Table 3: Time Consumption of Encryption/Decryption (ms) and (µs)

Key length	Encryption		Decryption		Total (Encryption & Decryption)	
	RSA	AES	RSA	AES	RSA	AES
1024 / 160	680 µs	230 µs	4	1	5	2
2048 / 224	870 µs	440 µs	16	2	17	3
3072 / 256	3	1	32	2	35	3
7680 / 384	6	2	559	2	565	4
15360 / 512	19	3	3456	4	3475	7

Although AES is faster than RSA in encryption, it is noticeable that the gap between ECC and RSA systems grows rapidly as the key sizes increase in key pair generation. Subsequently, the RSA system is much more time-consuming than the ECC system. Hence to obtain an efficient and higher security level, especially in the big data environment, the key sizes must expand and it is advisable to select the ECC-based system.

7.Statistical Analysis

Big Data has a characteristic of the variety, so it may contain images. Therefore, several tests were performed to encrypt the image using the proposed scheme. However, to resist statistical attacks, the encrypted images should possess certain random properties. Independence and uniformity are two properties that should be satisfied by any encryption scheme to ensure high resistance against statistical attacks. To prove the performance of the improved cryptosystem and verify these properties, a statistical analysis has been performed by calculating the correlation, SSIM, MSE, and histograms for the plain image and the encrypted image. Several images have been tested, and it has shown that the intensity values are good as illustrated in **Table 4**.

Table 4. Results of Differential Analysis

Image	Correlation		SSIM	MSE
	Original image	Encrypted image		
Lena	0.9959	0.0009	0.1061	0.3346
Airplane	0.9948	0.0004	0.1112	0.3347
Pepper	0.9966	0.0021	0.1067	0.03345
Tulips	0.9981	0.0061	0.0933	0.3384

The experimental result of the plain image and its corresponding cipher image and their histograms are shown in **Figure 2**. The figure shows the original plain image and its cipher image encrypted by the improving scheme. The result reveals that the improved scheme has reliable encryption and decryption effect. The histogram of the encrypted image is nearly uniformly distributed and significantly different from the respective histograms of the original images, which means that the improved algorithm has an excellent performance in resisting statistical attacks.

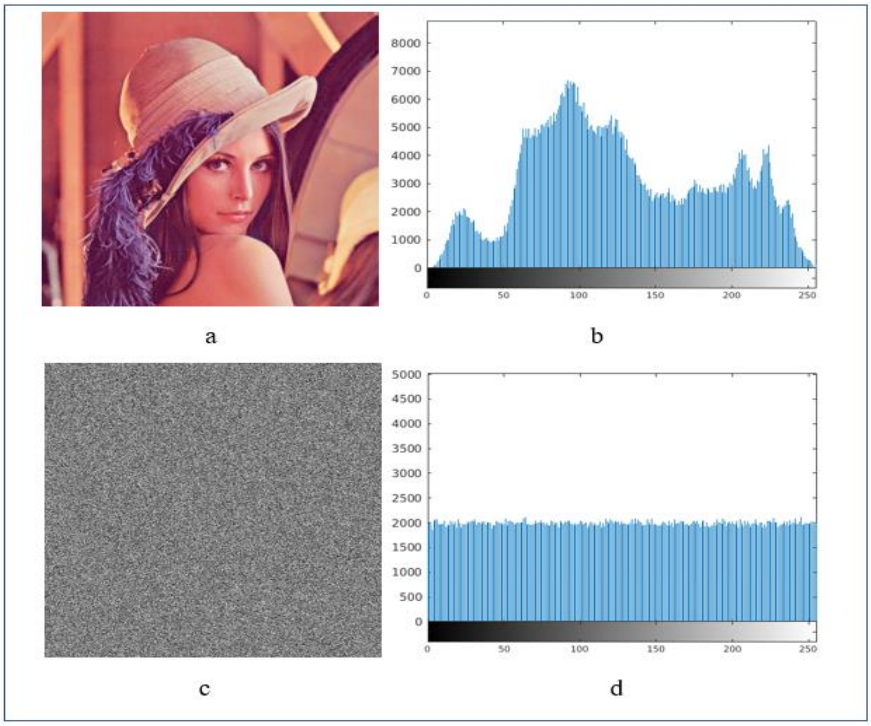

Figure 2. (a) Original Image (b) Histogram of Original Image (c) Encrypted Image (d) Histogram of Encrypted Image

8.Conclusion

An improving algorithm of Certificate-Based Encryption has been presented to using it in the Big Data environment. The system can handle the risks of the compromised CA and key escrow problem, as well as ensure the privacy of communications between clients because the key generation is done at the user level without any intervention from the certificate provider using ECDH, and hence the private key cannot be illegally obtained.

The combining of ECDSA as well ECDH will build a strong system that ensures secure communication among the entities. The key exchange process takes place after the completion of the authentication process, where the authentication is done by verifying the certificate. As a result, the key exchange operation is secured. The keys are then exchanged to generate a shared secret key, which is used as a key to encrypt and decrypt the data.

Consequently, this ensures that two-way data security each one completes the other. The certificate provider cannot decrypt the ciphertext, and at the same time, the user can not send the data without a verified certificate approved by the certificate provider. The proposed scheme tested on a big dataset of social networks. A comparison of the time consumption of the model has been tested and shown to be fast in key generation, encryption, and decryption. However, since the variety is a characteristic of big data, several tests were performed to encrypt the images using the proposed scheme and it has shown that the intensity values are acceptable.

References

1.Ahmed, H. E. H.; Kalash, H. M., ; Farag Allah, O. S. (2007). Encryption efficiency analysis and security evaluation of RC6 block cipher for digital images. **2007** *International Conference on Electrical Engineering,ICEE*. https://doi.org/10.1109/ICEE.2007.4287293.

2.A Shamir ,Identity Based Cryptosystem and Signature Schemes, *Advances in Cryptology Proceedings of CRYPTO*, **1984**, *84*, 7–53. Available at:

http://discovery.csc.ncsu.edu/Courses/csc774-S08/reading-assignments/shamir84.pdf.

3.Anoop, M. S. Elliptic Curve Cryptography – An Implementation Tutorial Elliptic Curve Cryptography An Implementation Guide. **2001** ,1–11. Available at:http://www.infosecwriters.com/text_resources/pdf/Elliptic_Curve_AnnopMS.pdf.

4.Cheng, L., Liu, F. ; Yao, D. D. Enterprise data breach: causes, challenges, prevention, and future directions, Wiley Interdisciplinary Reviews: *Data Mining and Knowledge Discovery*, **2017**, *7(5)*, 1–14. doi: 10.1002/widm.1211.

5.Galindo, D., Morillo, P. ; Ràfols, C. Improved certificate-based encryption in the standard model, *Journal of Systems and Software*, **2008**, *81(7)*, 1218–1226. doi: 10.1016/j.jss.2007.09.009.

6.Gentry, C. Certificate-Based Encryption and the Certificate Revocation Problem, International Conference on the Theory and Applications of Cryptographic Techniques: Advances in Cryptology — EUROCRYPT, **2003**, 272–293. doi: 10.1007/3-540-39200-9_17.

7.Goldwasser, S., Micali, S., ; Rivest, R. L. A digital signature scheme secure against adaptive chosen-message attacks, *SIAM Journal on Computing*, **1988**,*17(2)*, 281–308.

8.Daoud, A. O., Tsehayae, A. A.; Fayek, A. R. A guided evaluation of the impact of research and development partnerships on university, industry, and government. *Canadian Journal of Civil Engineering*, **2017**,*44(4)*,253–263.

9.Hwang, S. O. ; Le, M. H. Efficient certificate-based encryption and hierarchical certificate-based encryption schemes in the standard model, *Journal of Intelligent and Fuzzy Systems*, **2018**, *35(6)*, 5971–5981. doi: 10.3233/JIFS-169838.

10. Hyla T., Maćków W., Pejaś J. Implicit and Explicit Certificates-Based Encryption Scheme. In: Saeed K., Snášel V. (eds) Computer Information Systems and Industrial Management. CISIM 2015. Lecture Notes in Computer Science, vol 8838. Springer, Berlin, Heidelberg. **2014**, 651-666. doi:10.1007/978-3-662-45237-0_59.

11.Johnson, D.; Menezes, A. ; Vanstone, S. The Elliptic Curve Digital Signature Algorithm Validation System (ECDSAVS).**2004**, 56. Available at: http://cs.ucsb.edu/~koc/ccs130h/notes/ecdsa-cert.pdf.

12.Langford, Susan K. Threshold DSS signatures without a trusted party." *Annual International Cryptology Conference*. Springer, Berlin, Heidelberg, **1995**.

13.Lehmann, E. L.; Casella, G. Theory of Point Estimation , Second Edition *Springer Texts in Statistics In Design*. **1998**,*41*. https://doi.org/10.2307/1270597

14.Le, M. H., Kim, I. ; Hwang, S. O. 'Efficient certificate-based encryption schemes without pairing', *Security and Communication Networks*. **2016**, *9(18)*, 5376–5391. doi: 10.1002/sec.1703.

15.Liu, J. K. ; Zhou, J. Efficient certificate-based encryption in the standard model, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5229 LNCS, **2008**, 144–155. doi: 10.1007/978-3-540-85855-3_10.

16.Lu, Y. ; Li, J. Efficient certificate-based encryption scheme secure against key replacement attacks in the standard model, *Journal of Information Science and Engineering*, **2014**,*30(5)*, 1553–1568.

17.Lu, Y., Li, J. and Xiao, J. (2008) Generic construction of certificate-based encryption, *Proceedings of the 9th International Conference for Young Computer Scientists*, ICYCS

2008,1589–1594. doi: 10.1109/ICYCS.2008.11.
18.*S. Vanstone,* Responses to NIST's Proposal, *Communications of the ACM, 35, July* **1992**, *50-52 (communicated by John Anderson).*
19.Yang, J., ; Leskovec, J. Defining and evaluating network communities based on ground-truth. *Knowledge and Information Systems*, **2015**, *42*(1), 181-213.Retrievedfrom https://snap.stanford.edu/data/com-Friendster.html
20.Vaudenay, S. (2002) The security of DSA and ECDSA: Bypassing the standard elliptic curve certification scheme, *Public Key Cryptography — PKC* **2003**, 2567, 309–323.