iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** VIII **Month of publication:** August 2020

DOI: https://doi.org/10.22214/ijraset.2020.30929

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VIII Aug 2020- Available at www.ijraset.com

Security Application for Data Transfer using Cryptographic Protocols and Blockchain Concepts

Ankita Singh S[1], Anusha A Reddy[2], Apoorva N[3], Bharath B[4], Alpana Dahiya[5]
[1, 2, 3, 4]*Student,* [5]*Asst. Professor (Guide), Department of Computer Science & Engineering, MVJ college of Engineering, Karnataka, India*

Abstract: ***In the modern world, any kind of data is always important. The three prominent aspects to data communication are authenticity, integrity, and non-repudiation. While authenticity and non-repudiation can be ensured using cryptographic protocols, algorithms and network security layers, confidentiality assurance can be brought about using various stand-alone mechanisms; one of these is the Blockchain. A blockchain is a sequence/collection of blocks containing data. These blocks are immutable and as a whole the blockchain is managed by and distributed across a cluster of nodes in a network and not owned by any single entity. Each of these blocks are virtually related to each other using cryptographic principles. Because of this, changing a single block could result in a need to change all the blocks in the blockchain making the task practically impossible considering the time requirements. Such benefits of the blockchain can be used for achieving data confidentiality. The proposed project aims to design a portable software that can be deployed across organizations that rely on high security considerations while communicating media files viz., images, audio, video and documents making the communication at most secure due to the benefits of the cryptographic algorithms used and the services offered by the blockchain.***
Keywords: Cryptography, Blockchain, Secure file transfer

I. INTRODUCTION

Data confidentiality is about denying unlawful, unauthorized, unintentional use of data by an unauthenticated entity. Data that has low confidentiality requirements may simply be considered as public but data that has high confidentiality requirements needs to be protected to thwart identity theft, compromise of entire systems, legal damages and other severe consequences. Examples of such data are: social security numbers (SSN), e-mail passwords, bank and other account numbers, digital certificates etc. Cryptography is the go-to choice for protecting the confidentiality of data at rest and in motion. Even though it does introduce some complexities in computations and latency in communication, it can be used to make a trade-off between what is needed and what is to be achieved.

The most common mechanism that is used to communicate media files across two systems over the internet is by using the cloud frameworks offered by different cloud vendors like Google, Amazon, Microsoft, etc. But cloud often faces extensive disadvantages not only due to its operating procedures but also due to the fashion in which it is used. Since cloud-based solutions depend on the speed of the Internet upload and download, having a low latency can hinder you from accessing the data in real time. Also, there are still many areas around the globe where Internet connection is either difficult or not possible at all. Also, Customer support is not one of the stronger points of cloud storage vendors. Cloud storage providers usually instruct clients to take a closer look at the FAQ or follow online forums instead of offering a one on one assistance to the customer to simplify the customer's experience. After migrating the data to a third-party cloud storage provider, there is a bit of an issue concerning who owns the information. Is it the customer as a client, or the cloud storage provider?

Another popular alternative to file sharing is the FTP way. But FTP is inherently a non-secure way to transfer any kind of data. When a file is sent using this protocol, the data is shared in plain text, which means anyone can access this information with little to no effort. To solve the problem of security vulnerabilities, consumers turn to hosted FTP solutions managed by vendors. Unfortunately, not all vendors provide the needed security and each offers their own set of features and many lack the features like access controls, security, usability, pricing options, etc. FTP with inadequate security could leave a business at risk of noncompliance fines. Compliance regulations like HIPAA, ITAR, PCI-DSS, SOX, or GLBA are often not fulfilled by FTP solutions.

The problems unresolved by cloud and FTP is being tackled by the proposed project by performing file sharing without the involvement of a third party and with cryptographic protection provided to the transmitted data. The elimination of the third-party provider is done by making the file sharing a peer-to-peer process using similar mechanisms that is seen in Blockchain networks. The masking and hiding of data to outside systems is achieved by cryptographic protocols. Some of them are described below.

To enhance the capabilities of cryptography, we utilize the blockchain technologies originally implemented for the cryptocurrency, bitcoin. Information contained in a blockchain exists as a shared, continually revised and reconciled database and since it is not

stored in one location, records are truly public and easily verifiable. There is no centralized version of this information which can be corrupted by a hacker. The three pillars of blockchain are described below:

1) *Decentralization:* Information is not owned/stored by a single entity but rather everyone in the network owns the information. This means that A can communicate with B directly without relying on a third-party C. The Bitcoin uses this ideology where in A is alone in-charge of the money it owns and A can send Bitcoins to any other node without going through a bank.
2) *Transparency:* Although the data in blockchain is publicly transparent, it does not mean that privacy is not provided by the blockchain system. In case of bitcoin, bitcoin transactions are recorded in the blockchain. These transactions include the identity of the transacting parties. However, the identities are not stored in the clear but are hidden using complex cryptography and represented only by their public address.
3) *Immutability:* Once data is entered into blockchain, it can't be manipulated. This is because of the cryptographic hash function which takes an input string of any length and gives an output of a fixed length. The blockchain can be viewed as a linked list consisting of data part and a hash pointer. The hash pointer is the hash of the data inside the previous block hence creating the chain. The blockchain is maintained by a peer-to-peer network that partitions its entire workload between the peers. The backbone of the proposed system is made of many diverse cryptographic algorithms. These algorithms are used for important tasks like data encryption, authentication and digital signature.

An important part of maintaining a blockchain is the mining process. Miners constantly search for a new block to be added to the blockchain. When a block has been generated by a node and is to be added to the blockchain the hash of the contents of the new block is produced. A nonce (random number) is generated and appended to the hash. The resulting new string is hashed again. The final hash is then compared to the difficulty target and checked to see if it's actually less than that. If not, then the nonce is changed and the process is repeated again. If yes, then the block is added to the chain and the public ledger is updated of the addition. The miners responsible for this are then rewarded with bitcoins.

II. METHODOLOGIES

A. *Diffie-Hellman Key Exchange*

The main purpose of the Diffie-Hellman key exchange is to securely establish shared secrets across systems that can be used to derive keys. Let us assume Alan wishes to establish a shared secret with Bill.

1) Alan and Bill agree on a finite cyclic group G of order n and a generating element g in G.
2) Alan picks a random natural number a, where 1 < a < n, and sends g^a to Bill.
3) Bill picks a random natural number b, which is also 1 < b < n, and sends g^b to Alan.
4) Alan computes (g^b)^a. Bill computes (g^a)^b.
5) Both Alan and Bill are now in possession of the group element g^(a*b), which can serve as the shared secret key.

B. *RSA Cryptosystem*

The RSA algorithm involves four stages: key generation, key distribution, encryption and decryption.

C. *Key Generation*

The generation of the keys for the RSA algorithm is done in the following way:

1) Choose two distinct prime numbers p and q that are to be kept secret.
2) Compute n = pq. n is the modulus for the public and private keys. Its length is the key length.
3) Compute λ(n), where λ is Carmichael's totient function and it is calculated as λ(n) = lcm(p − 1, q − 1). λ(n) is kept secret.
4) Choose an integer e such that 1 < e < λ(n) and such that gcd(e, λ(n)) = 1 making e and λ(n) coprime.
5) Calculate d as d ≡ e^(-1) (mod λ(n)); i.e., d is the modular multiplicative inverse of e modulo λ(n).
6) The public key consists of the modulus n and the public exponent e. The private key is d, which must be kept secret. p, q and λ(n) must also be kept secret as they can be used to calculate d.

D. *Key Distribution*

Suppose that Bill wants to send information to Alan. If they use RSA, Bill must know Alan's public key to encrypt the message and Alan must use his private key to decrypt the message. If Bill is to send his encrypted messages, Alan has to transmit his public key (n, e) to Bill via a reliable route.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VIII Aug 2020- Available at www.ijraset.com

E. Encryption

After Bill obtains Alan's public key, he can send a message M to Alan.

To do it, he first turns M into an integer m such that $0 \leq m < n$. He then computes the ciphertext c, using Alan's public key e, corresponding to $me = c \pmod n$

Bill then transmits c to Alan.

F. Decryption

Alan can recover m from c by using his private key exponent d by computing

$cd = m \pmod n$

Given m, he can get back the original message M by reversing the padding scheme.

3. Vigenere Cipher: Vigenere Cipher is a method of encrypting alphabetic text using simple keys. It uses a simple form of polyalphabetic substitution. A polyalphabetic cipher is a cipher based on substitution, using multiple substitution keys. Vigenere can also be described algebraically. If the letters A–Z are taken to be the numbers 0–25 and addition is performed modulo 26, Vigenere encryption E using the key K can be written as

$C_i = (M_i + K_i) \bmod 26$

and decryption D using the key K as

$M_i = (C_i - K_i + 26) \bmod 26$

in which $M=M_1...M_n$ is the message, $C=C_{1....}C_n$ is the ciphertext and $K= K_1....K_n$ is the key obtained by repeating the keyword.

III.PROPOSED SYSTEM

A. Implementation

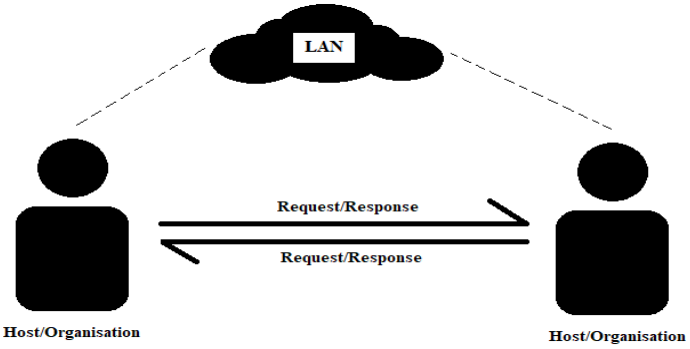


Fig 2: System overview

The proposed software makes way for a communication between two entities. These entities can be either individual hosts or organizations that have installed the software in their systems. The communication is governed by a LAN. Any data(documents or media) may be transferred between the two entities by having a request-response protocol acting between them. The entities are required to be within the same LAN network such that their IP addresses are accessible to each other.

B. System model

The system consists of the following two entities:

1) Sender: Sender is a person or organization that owns a one or many files to share.

2) Receiver: Receiver is a data client that need a file and request for it from the Sender.

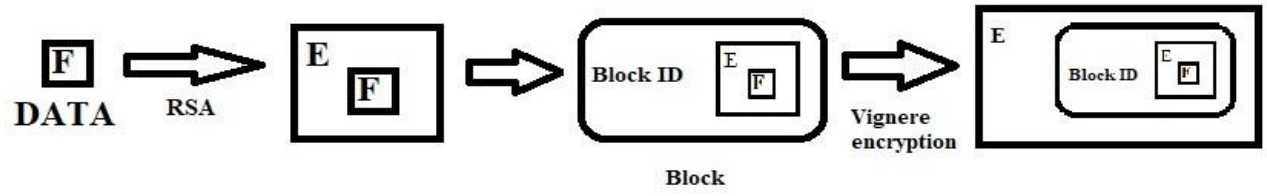


Fig 1: System model

C. Algorithm (Encryption)

1) Input: File to be sent
a) Convert input file into .txt file.
b) Break the file down into chunks.
c) Encrypt the chunks 1…N using RSA public key.
d) Use Diffie-Hellman secret to derive Block-1 encryption key using sha256 and HKDF.
e) Use chunks 2…N to derive Block-2…N encryption keys using sha256 and HKDF.
f) Encrypt blocks using corresponding keys.
g) Send the encrypted blocks.
2) Output: Encrypted blocks are sent to receiver side.

D. Function Implementations

The system model consists of processes/protocols implemented as shown below:

1) *diffieHellman(s) → secret:* This algorithm is run by the Sender. The input is the socket object s. the output is the secret that Diffie-Hellman is required to generate. The sender would create a generator g and prime number p. both g and p are sent over to receiver through a secured channel. The sender generates prime number a. The receiver generates prime number b. Once the key preliminaries are established, the sender sends g^a mod p and receives g^b mod p and receiver does vice-versa. the sender and receiver calculate (g^b mod p) ^a and (g^a mod p)^b.
2) *rsaPublicKey(s) → (e, n):* the algorithm is run by the receiver to generate and end the public key to sender. The input taken is the socket s. the output is a tuple of public key e and modulus n. the algorithm generates two prime numbers p and q, calculates modulus and totient function, phi. Using these variables it then calculates the public key e. and ultimately transmits a tuple of(e, n) to sender.
3) *deriveSessionkey(hd) → key:* The sender and receiver can both run their algorithm to derive session keys for individual blocks. The input is the hexdigest of the hash value hd. The output is the hexdigest of the derived session key i.e key. The procedure employed to derive the key is HKDF. It extracts a pseudorandom key (PRK) using an HMAC hash function (e.g. HMAC-SHA256) on an optional salt and any potentially weak input key material (IKM) (acting as data). Then it generates output key material (OKM) of desired length by generating repeatedly, PRK-keyed hash-blocks and then appending them to the OKM and truncating to desired length. Here length is a pre-determined parameter.
4) *encryptBlock(blockData, S_k) → blockCipher:* the sender runs this algorithm to encrypt individual blocks. The input is data stored inside the block, blockdata and session key S_k. The output is the cipher, blockCipher, which is a string of characters. The algorithm builds a dictionary consisting of the English alphabet in both upper and lower cases along with decimal numbers and printable punctuation symbols. S_k is used as the multidigit key for the Vigenere encryption and the alphabet used for the encryption is the dictionary built by the algorithm.

IV. CONCLUSIONS

An implementation of a Blockchain-based methodology to transfer files is demonstrated. Any type of file can be transferred by converting it into a text file using Base64 encoding. The raw text encoding is encrypted using RSA due its benefits provided in terms of the co-factoring needed to break it.

Using Diffie-Hellman also poses the discrete logarithmic challenge when it comes to solving for prime numbers. The blocks act just the same as those found in a Blockchain ledger being interrelated by hash values. Thus, the blocks either work together to build the whole file or not at all.

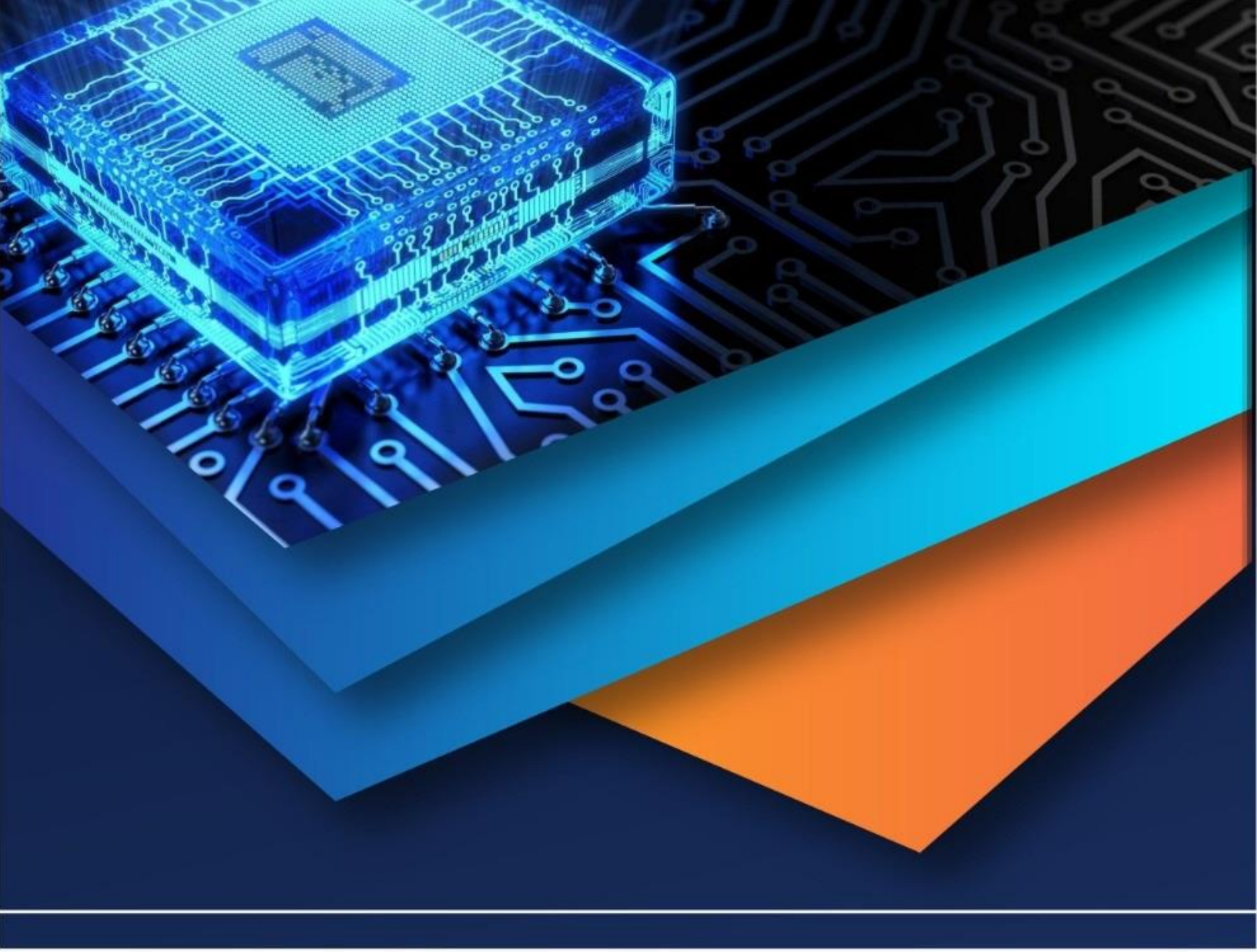
V. FUTURE WORK

The current implementation can be further enhanced by allowing systems to connect over the internet via IP addresses or other mechanisms.

Also, the project is also currently being made into a mobile application on Android platform capable of running on Android versions of 5.1 and higher. We also plan to develop the application in iOS platforms in the future. The time taken to transfer files(documents/media) can be also improved and made faster by improvements to the algorithm itself.

REFERENCES

[1] Alevtina Dubovitskaya, Zhigang Xu, Samuel Ryu, Michael Schumacher, Fuheng Wang, “Secure and Trustable Electronic Medical Records Sharing using Blockchain”, AMIA, 2018.
[2] Chainworks Digital LLP, “Know Your Customer - Decentralized and Secure Sharing Protocol on Quorum”, 2019.
[3] Deepak K. Tosh, Sachin Hetty, Peter Foytik, Charles A. Kamhoua, “CloudPoS: A Proof-of-Stake Consensus Design for Blockchain Integrated”, IEEE, 2018.
[4] Jingwei Liu, Xiaolu Li, Hongli Zhang, Xiaojiang Du, Mohsen Guizani, “BPDS- Blockchain based privacy-preserving data sharing for electronic medical records”, arXiv, 2018.
[5] Ryusei Fuji, Shotaro Usuzaki, Kentaro Aburada, Heraaki Yamaba, “Investigation on Sharing Signatures of Suspected Malware Files Using Blockchain Technology”, IMECS, March 2019.
[6] S.M.K.V. Pramod kumar, Kiran kumar, Sai Krishna R, P.S.G Aruna Sri, “Incorporation of Blockchain In Student Management System”, IJITEE, April 2019.
[7] Shangiping Wang, Yinglong Zhang, Yaling Zhang, “A Blockchain-Based Framework for Data Sharing with Fine-Grained Access Control in Decentralized Storage Systems”, IEEE Access, 2018.
[8] Sri Balaji, Vignesh Mohan, Soundarya, “Secure and Decentralized File Transfer Application Using Blockchain”, TROI, 2017.
[9] Yongle Chen, Hui Li, Kejiao Li and Jiyang Zhang, “An improved P2P File System Scheme based on IPFS and Blockchain”, IEEE, 2017.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY