Dual Authentication for Payment Request Verification Over Cloud using Bilinear Dual Authentication Payments Transaction Protocol

A. Saranya[1]
Research Scholar, Department of Computer Science and Engineering
SRM Institute of Science and Technology
Kattankulathur, Chengalpattu, Chennai, Tamil Nadu, India

R. Naresh[2]*
Associate Professor, Department of Networking and Communications
SRM Institute of Science and Technology
Kattankulathur, Chengalpattu, Chennai, Tamil Nadu, India

***Abstract*—There has been a recent explosion in the number of mobile network payment gateways that enable users to access services through a variety of devices. Mobile payment gateway security is complicated by a number of difficult-to-solve issues. As digital technology has progressed over the last decade, mobile payment mechanisms have gained a lot of interest. In the internet industry, these standards might have a significant impact on service quality. However, the most important aspect to consider when using these systems is their accountability, which assures confidence between the parties engaged in the financial transactions. Mobile payments may be easy, quick and secure. On the other hand, they may be rather pricey and are still susceptible to problems caused by technology. Specifically, mobile payments won't be able to go through at all if there are any problems with the host phone. For this reason, in this article a mobile payment mechanism that uses secure bilinear dual authentication. Using Bullet hash Maximum distance separable (MDS) and the mutate Hellman algorithm, our payment protocol incorporates all of the essential security characteristics to establish confidence between the parties. To put it another way, accountability is assured by mutual authentication and non-repudiation. Conflicts that may emerge in the course of a payment transaction may be resolved using our strategy. Scyther is used to test our suggested protocol's empirical performance.**

Keywords*—Mobile payment; transaction protocol; bullet hash maximum distance separable; mutate Hellman algorithm

I. INTRODUCTION

Mobile smart devices (such as smartphones or laptops) have become more common in everyday life as a result of the development and widespread usage of mobile communication technology [1]. This results in an ever-increasing amount of online service requests. Online services such as Ali Pay, Apple Pay, We Chat Pay and so on rely on mobile payments, which are attracting a lot of attention. These online transaction programs let customers to purchase a wide range of goods and services from wherever they have internet access. A user's private identifying information is often made available to retailers when an online transaction is initiated in order to verify the transaction's authenticity. The unreliability and avarice of merchants may lead them to offer items that the user does not need or merely to sell the identity of the user for economic gain to other parties. There must be an ability to validate a transaction message's legality and validity, so that the merchant may ensure that products or services are given by the proper user. [2] Verification of transaction messages may also prevent users from claiming that they didn't acquire the products and services they claimed to have purchased. Many cryptographic primitive-based protocols for mobile payments have been developed to meet these security requirements. Their protocols meet security standards such as user anonymity and unforgivability. The identity of any merchant or opponent cannot be linked to a transaction message when a protocol maintains anonymity for the users involved [3]. Unforgeability, on the other hand, implies that the sender of a message can be identified and that anybody attempting to forge the transaction message of another party will be caught [4]. A mobile payment system must take efficiency into consideration in addition to security considerations. As a result, it is important that a payment protocol only need a minimal amount of processing power and storage space to be implemented on low-powered gadgets. Traditional transaction protocols provide user public key certificates through a public key infrastructure (PKI) [5, 6]. Verifying the authenticity of a public key may be done with the help of a trusted certificate authority. Because of the need for extra resources to manage certificate revocations, distribution, and storage, the use of PKI significantly raises the costs of both communicating and storing data. As a result, there is a contradiction between PKI and the capabilities of mobile devices with limited processing and storage. It is still difficult to build a mobile transaction protocol with minimal resource requirements in terms of computation, network traffic, and data storage. Addressing the aforementioned issue, a novel mobile payment system was created that simultaneously guarantees secrecy, immutability, and minimal resource usage. The following is a brief summary of its most important contributions:

1) First, the dual authentication payment transaction method was provided.

2) A mobile payment mechanism based on our suggested approach is demonstrated. Furthermore, Pay Platform serves as a trusted proxy for users when they want to securely communicate with Merchant Server. As a result, users can rest easy knowing that they won't be sending or receiving private messages from merchants. Since most calculation takes place

on Pay Platform, user resource utilization is lowered as well. Unforgeability is ensured by the use of certificate less public key cryptography and signatures on every piece of transaction data.

3) It is important to maintain the Pay Platform and Merchant Server as lightweight as possible, despite the fact that they must do computations for each transaction, in order to solve the scalability problem. When processing a payment on the Pay Platform or the Merchant Server, the signature verification process is by far the most time-consuming step.

4) The protocol was implemented to test the using the Bullet hash MDS, a mutated version of the Hellman algorithm, and various mobile payment protocols. Comparison reveals that our protocol is viable and efficient in terms of secure payments.

Rest of the paper has been divided into sections. Section II focuses on other studies in this field. The problem statement is presented in Section III. Section IV demonstrates the suggested technique's terminology in action. Section V focuses on experiments and implementation. Towards the end of the article, the conclusion is reached.

II. Related Work

Mobile payment systems have been the focus of a number of researches in recent years aimed at strengthening their security. Many of these efforts have been directed to developing foundations for the establishment of new electronic payment systems, as well. Username and passwords, symmetric and non-symmetric and elliptic curve encryption, smart cards, 2D bar codes, and biometric technologies have all been tried for electronic payment system authentication. All sorts of authentication techniques and protocols are based on these notions. Symmetric and asymmetric signatures fail in M-commerce despite being frequently utilized for authentication. In [7], mobile payment solutions based on durable certificate-less signatures and bilinear pairing are proposed. [7] To make the suggested mobile payment system acceptable for mobile devices with low processing capacity, they smartly improve it. Security and performance testing of the suggested mobile payment system on the Raspberry PI have demonstrated its feasibility. Anonymous and untraceable payments have never previously been feasible on a mobile device according to a novel payment technique explained in [8]. Because mobile devices have limited computing capacity, the proposed protocol depends on a Pay Platform to handle the majority of the computational burden (which is almost usually equipped with lots of processing power) (which is almost always equipped with plenty of processing power). Batch-verification has been implemented to lessen the overhead for millions of users on the Pay Platform and Merchant Server because the Pay Platform and Merchant Server must run calculations for each transaction. In terms of cloud security and privacy, one of the most pressing challenges is whether or not sensitive information may be accessed by other parties. The researchers [9] have presented cloud service providers with an intelligent encryption solution that secures cloud service providers' access to the incomplete data. Dispersed cloud servers are recommended for data storage after the data file has been partitioned. SA-EDS is a notion and an algorithm that underpins the Secure Efficient Data Distribution (SED2) Method, the EDCon Methodology, and the Alternative Data Distribution (AD2) Algorithm. ZeroMT is a method presented in [10] that enables several simultaneous covert balance transfers. This method ensures that the balances in all accounts and the amounts transferred remain private." "Since all transfers are contained within a single transaction, there are fewer transactions to verify on the main chain. Off-chain, they are the standard building block for a multi-transfer transaction that is undetectable by unauthorised parties but can be validated by smart contract infrastructures. NFC devices and payment terminals connect with each other using this protocol, which is based on the EMV standard [11]. EMV's weaknesses will be solved by adding an extra layer of security and beefing up the core EMV communications, according to the protocol. Due to the lack of a reliable third party, security features such as the detection of double spending and the prevention of brute force attacks are still in development. In order to make secure mobile payments, NFC-enabled cellphones may be utilised, as stated in [12]. Abughazalah's protocol has the potential to address issues of privacy and security. In order to ensure the security of this protocol, a one-time password (OTP) system was implemented. The OTP mechanism is preinstalled in NFC-enabled smartphones. However, without a trusted third party, this protocol lacks essential security features like the ability to prevent double spending and ensure all participants are treated fairly. According to [13], mobile vouchers for NFC-enabled phones are safe to use. The suggested strategy focuses largely on the collection and redemption of loyalty points. However, there are still several potential security holes in this system, including those related to message privacy, mutual authentication, and detecting duplicate spending. This results in an inherently unfair implementation of the protocol. An EMV-compliant near-field communication (NFC) mobile payment system was presented by the author in [14]. Data and digital signatures are secured by cryptographic methods like symmetric key encryption and public key encryption in their proposed protocol. Unfortunately, this protocol is vulnerable to a wide range of attacks despite the fact that it is not fair and cannot rely on a neutral third party to ensure its security. In [15], the author presented a secure contactless NFC payment system using NFC bank cards and an internet connection with a respectable organization. Client payment devices limited to NFC bank cards that do not need Wi-Fi or 4G. Due to the absence of a reliable third party, this protocol is not yet capable of detecting duplicate spending or protecting against brute-force attacks, and it is therefore not yet fair. An NFC mobile payment system with cloud-based security was shown in [16]. The usage of NFC radio waves in a public setting makes it more secure than the EMV standard. Other security flaws, such the ability to spot double spending, may be exploited even in the absence of a third party that can be trusted. The au's anonymous mobile payment system, which was introduced in [17], has increased the safety of mobile commerce data. They assert that their protocol satisfies all of the necessary conditions for non-repudiation, anonymity, and unlinkability. However, there is no safeguarding against message corruption or brute-force attacks in this protocol [18-21]. The trustworthy third party's knowledge of sensitive transaction data like

payments and invoices also poses a security risk. Using key distribution, the author of this research presents [22] a cloud-based efficient authentication system for mobile payments. Our innovative certificate-less proxy re-signing approach for mobile payments not only protects your privacy, but also simplifies your data storage needs. Using the Secure Authentication Protocol, [23] suggests a mobile payment system (SAP). By using cryptographic methods, it was able to create a reliable solution that can identify fake servers and clients. While the use of mobile devices in the payment industry continues to rise, the proposed method ensures the safety of user accounts and individual privacy. Smart mobile device makers and mobile data users both have a role to play in resolving this problem so that mobile payments may continue to be used in public communication networks that are not always secure. They provide a secure business strategy for mobile payments in the e-healthcare application by making use of key distribution cryptographic techniques. In [24], a novel, effective trust model for secure, dynamic group communication across distant networks is presented. The concept of a hierarchical attribute-based encryption was introduced by the author in [25]. A single encryption key may secure a whole directory's worth of data. The computational and storage requirements of their attribute-based hierarchical file encryption method are less than those of prior research. A novel index model, the data vector (DV) tree, based on a crossover genetic process, was developed with the help of soft computing. The file's term and inverse document frequencies, together with any other features included in the file, are used to build the DV tree. Their new key generation, encryption, and decryption tool is an extra bonus for anybody using their data. The authors of [26] proposed a machine learning algorithm to identify malicious uniform resource locators by combining URL lexical selections, payload size, and python supply parameters. A Chaotic Hopfield Neural Network combined with an adaptive encoding approach may provide a more secure model for keeping private information; the proposed approach enhances the safety of a shared key among any number of nodes [27-32].

A. Research Gap

Consequently, the existing solutions for mobile payment protocols lacked both substantial data security and fairness. There is no credible third party under the constraints of such procedures. To avoid wasting time and money in the event of a disagreement, no one gathers and records all transactional data. This means that the processes at play cannot ensure a fair outcome. As a result, there is potential for a variety of attacks, including physical force, man-in-the-middle, and double spending. In this work, a strategy for resolving these issues is demonstrated. That is to say, the proposed protocol simultaneously guarantees both high levels of fairness and critical data security. The offline session key generation and hashing features are used for further security and mobility. Basically, it was impossible to stop repetition or brute-force attacks by never reusing the session keys that are created for each and every transaction.

III. Problem Statement

No one will trust m-commerce until there is a secure method of exchanging business information and conducting electronic financial transactions over mobile networks. Secure connections and transfers are essential for M-payment systems, which send sensitive data. As a result, high levels of perceived and technological security are required for m-payment systems to be widely used and widely accepted by customers. M-commerce has seen the implementation of a number of different mobile security and payment mechanisms. Cryptography techniques are essential to satisfy the above-described transaction security standards. The security of mobile payments done via networks with little or no physical protection is greatly enhanced by these devices.

IV. Proposed Work

Client, Merchant, Acquirer, Issuer, and Payment Gateway are all parts of the BDAPTP architecture was developed (which is in turn based on the Client Centric model) (all of which were described earlier). Below are three examples of simple Payment transactions that illustrate the Client Centric approach in action.

- When a customer completes a purchase, the money is transferred to the shop owner.
- With Value Subtraction, the Client's funds are deducted by the Payment Gateway (on behalf of the Issuer).
- The Acquirer's Payment Gateway sends funds to the Merchant's Value Claim account. When using a Payment Gateway, you may get in touch with an Acquirer straight away (an entity that provides infrastructure for a Merchant to take credit card and other kinds of electronic payment). The suggested method has no need for communication between the merchant and the acquirer. A wireless or cellular network provided by a mobile phone provider is used to create an Internet connection between the Client and the Payment Gateway. The overall illustration of the suggested methodology was depicted in the Fig. 1.

A. BDAPT Protocol

There are a variety of processes for payment transaction depicted in Fig. 2. An individual identification number was assigned to each mobile phone. If an IMEI number is placed on a blacklist by a wireless network operator, GSM, it is used to identify legitimate devices on the network and to prevent ongoing usage of a phone that has been blacklisted from using the network. Different aspects of the suggested mobile payment security were identified in this article. The stages involved are:

1) Registration phase: Users and merchants must register on the payment gateway in order to get legitimate credentials, such as a mobile PIN for login and the encryption key, which must be generated.

2) Transaction phase: To make a payment, a user must first send a payment request to the merchant, which the merchant must confirm by verifying the user's payment information and personal information.

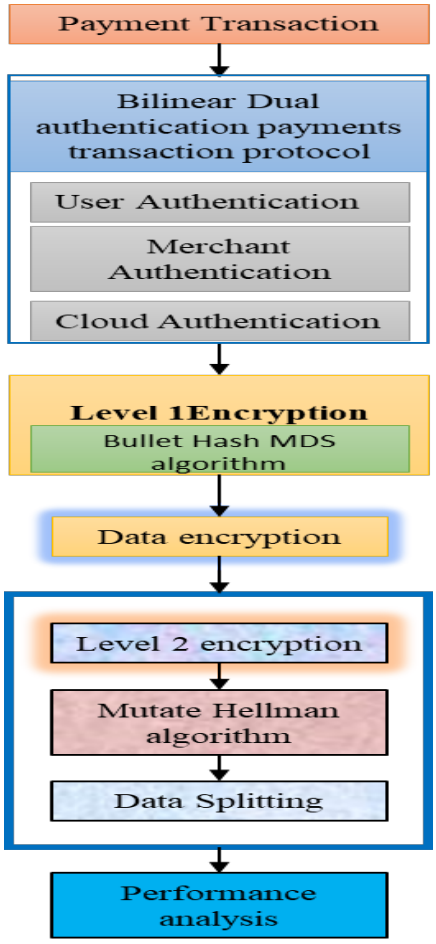

Fig. 1. Schematic Representation of the Suggested Methodology.

3) Authentication phase: The Issuer and Acquirer calculate the trust value to authenticate the payment credentials and approve the payment via the confirmation from the merchant module, which is then executed on the payment gateway.

It shows the process of the merchant server authentication using the proposed protocol. Merchant and admin servers must agree to the process of generating and disseminating a secret key for data encryption and decryption during authentication in order for the suggested method to operate. The Merchant Server employs "bank number" and a secret key to authenticate the admin server, creating a basic ciphertext when the authentication process is complete. The cipher text formula for merchant server authentication is:

$$\text{Merchant Server Authentication (MSA)} = \text{bankno (xor) trust value} \quad (1)$$

In order to avoid a session hijacking on the buyer's end, each authenticating demand should be issued a distinct private key.

$$\text{Admin Server Authentication (ASA)} = \text{MSA (xor) trust value} \quad (2)$$

$$\text{Admin server trust value (ASV)} = \left(g^{\text{bankno}}\right) \quad (3)$$

$$\text{Merchant share value(MSV)} = \left(g^{H(\text{bankno}_{no})}.g\right)^{\text{bankno}} \quad (4)$$

Whenever a client sends a payment, ask for the customer's transaction number in the cloud. Processing the request data requires the use of a trapdoor-creating method for the simple reason that if the payment requests were transmitted over the plain text, we run the risk of them being tracked and attacked. As a result, enhancing security and preventing the attack may be accomplished via the trapdoor design process.

$$\text{Patient Number Request (PNR)} = \left(g^{H(\text{bankno}_{no}).pn}, g^{pn}\right) \quad (5)$$

where $PN1 = g^{H(\text{bankno}_{no}).pn}$ and $PN2 = g^{pn}$

The cloud should now check and validate the merchant data and the mobile user request, as shown below, after it has received the whole request, trusted cloud exchanged between merchant server and mobile user across a distribution channel.

$$\text{Cloud payment request matching} = e\,(PNR, MSV) \quad (6)$$

$$= e\left(\left(g^{H(\text{bankno}_{no})}.g\right)^{\text{bankno}}, g^{pn}\right) \quad (7)$$

$$= e(g,g)^{(H(\text{bankno}_{no})+1).\text{bankno}.pn} \quad (8)$$

$$= e(g,g)^{H(\text{bankno}_{no}).\text{bankno}.pn}.e(g,g)^{\text{bankno}.pn} \quad (9)$$

$$= e\left(g^{H(\text{bankno}_{no}).pn}, g^{\text{bankno}}\right).e\left(g^{\text{bankno}}, g^{pn}\right) \quad (10)$$

$$e\,(PNR, MSV) = e(PN1, ASV).e(ASV, PN2) \quad (11)$$

A user's User Id is required before they may initiate a cloud transaction. The system will direct the user to the registration page if they have not previously done so. Once a user's id is found in the bank's database, the next step is to have them input a password. It is verified and if found to be inaccurate, the user is led back through the process, or if it is correct the user will be sent to a new step in which the system will ask for a unique identification (UID). For 24 hours, the user's account will be suspended if the incorrect UID is entered. If the user's UID and QR Code are matched, an OTP will be sent to the user's registered cellphone number; if the user's UID and QR Code are not matched, the user's account will be blocked for 24 hours. Once the OTP has been entered, the following step is to verify that the OTP entered is accurate. If the OTP is accurate, the key is kept secret throughout the encryption step. For 24 hours, the account will be blocked if the OTP entered is wrong. If the OTP is entered correctly three times, the system will send you to the right OTP entry procedure. All input is encrypted using the Bullet hash MD5 algorithm, a public key cryptosystem that provides a hash key. To further increase the degree of security in the suggested technique, the asymmetric public key cryptosystem modify Hellman algorithm was used to generate the hash key. Data is subsequently divided into 128-bit XOR operations and sent to the server for further processing.

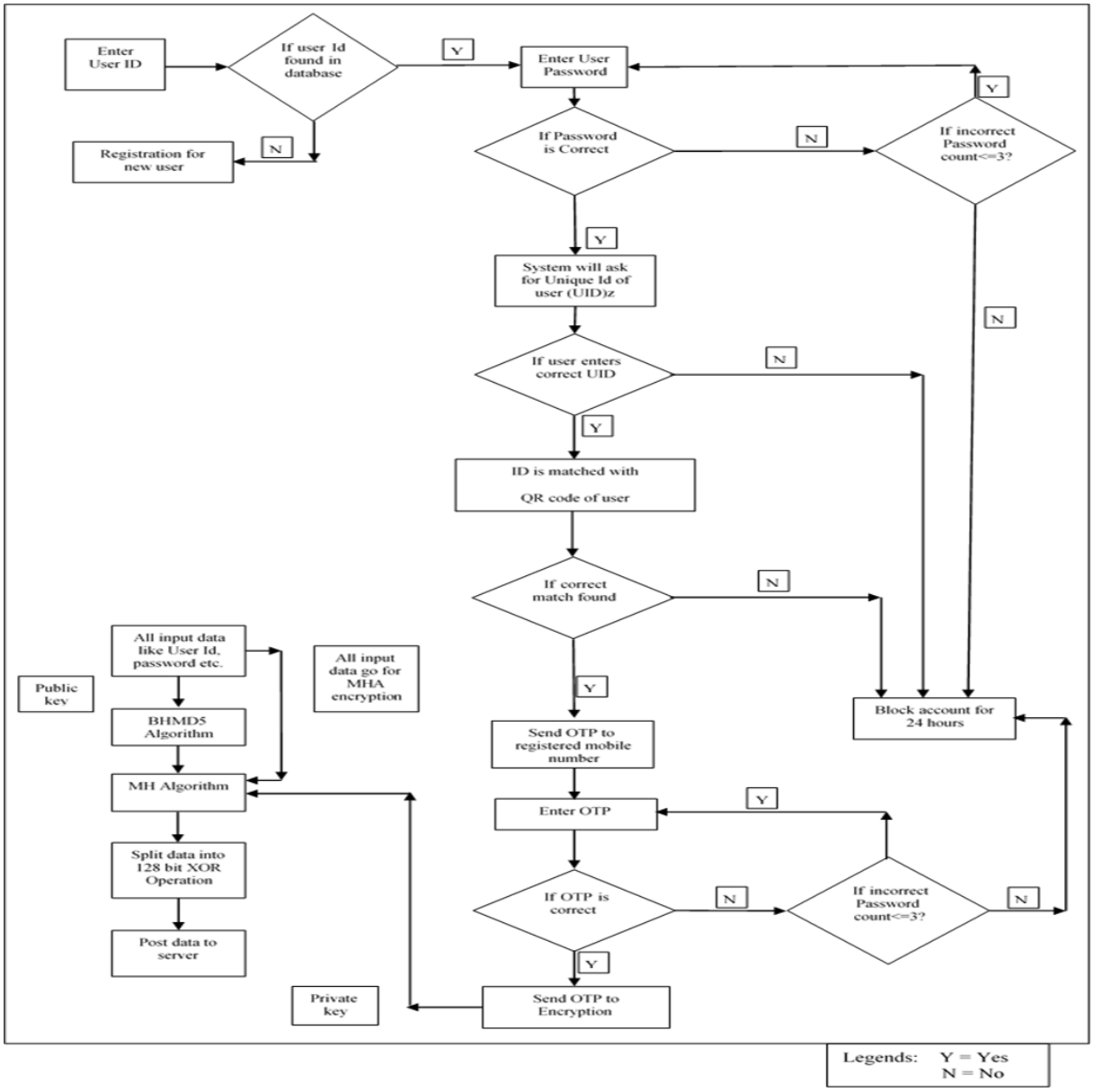

Fig. 2. Process of Transaction.

B. Units Cryptographic Transaction

1) Bullet hash MD5 algorithm: Integrity security is mostly provided via hash functions. Authentication is also provided when they are used in conjunction with digital signature and message authentication code (MAC) techniques. As an example of an important family of hash functions, consider the SHA-2 family, which has the same basic functional structure but varies in internal operations, message length and security bits.

A message is fed into one of these algorithms, which performs repetitive, one-way operations to produce a digest of the message. Blocks of 512-bit messages and 256-bit hash values expressed as eight 32-bit words are processed 64 times by the BHMD5 algorithm (C, D, ..., I). The hash message is 256-bits in length.

The following are some possible descriptions for the system:

$$\begin{cases} \dot{z} = l(t - z) \\ \dot{t} = xz - t - zm \\ \dot{m} = zt - vm \end{cases} \quad (12)$$

Where, z, t, and m represent the current state of the system, whereas l, v and x represent its parameters. A system enters chaos when a, b, and c are all equal to 10.

The BHMD5 was utilized to generate a 256-bit secret key J in the suggested cryptosystem. No matter how minor a difference there is between two photos, their hash values will be different. Thus, a 2256-complexity cryptosystem can withstand a brute-force attack without being breached. Accordingly, J may be represented as follows: 256-bit secret key split into 8-bit blocks (j_u).

$$J = j_1, j_2, j_3, \dots, j_{32} \quad (13)$$

The initial values can be obtained as follows.

$$z_0 = z_0' + \frac{(j_1 \oplus j_2 \oplus j_3 \oplus \dots \oplus j_{11})}{256} \quad (14)$$

$$t_0 = t_0' + \frac{(j_{12} \oplus j_{13} \oplus j_{14} \oplus \dots \oplus j_{22})}{256} \quad (15)$$

$$m_0 = m_0' + \frac{(j_{23} \oplus j_{24} \oplus j_{25} \oplus \dots \oplus j_{33})}{256} \quad (16)$$

Where, z_0', t_0' andm_0' are the initial given values.

Four elements make up the suggested encryption algorithm. Transaction data is first encrypted using the hash technique to create J and the Lorenz system's starting values. Encryption rules are then applied to the original data sequence J. Finally, decipher the series of numbers. Encrypted data is the product of the process.

Step 1: For each row and each column, enter the data in the form of O(n,b), where n is the number of rows.

Step 2: Generate the starting values of J and the key sequence (z_0', t_0',m_0') of the Lorenz system.

Step 3: The matrix N_j (n, b ×8) may be generated by repeating the binary sequence $J_{v,r}$ times where $r = \frac{n \times b \times 8}{32}$. Encode N_j with the same encoding rule and N_{jw} was obtained.

Step 4: According to XOR operation, Oe' = Oe XOR N_{jw}, Of'= OfXOR N_{je} and Ov' = OvXOR N_{jw}.

Step 5: The chaotic sequences are generated as z_b, t_bandm_b and its length is n× b× 4 by using the Lorenz system will having the initial values z_0', t_0' andm_0'.

Step 6: Prepare the chaotic sequences z_b, t_bandm_b as follows:

$$\begin{cases} (kz, dz) = sort(z) \\ (kt, dt) = sort(t) \\ (km, dm) = sort(m) \end{cases} \quad (17)$$

where (•, •) = sort (•) a new sequence after rising to an index value z is the new sequence, and the index values for this new sequence are the index values for this new sequence.

Step 7: The binary matrices should be converted $Oe', Of'$and$Ov'$to three vectors Ce (n× b× 4), Cf (n× b× 4) and Cv(n× b× 4), respectively.

$$\begin{cases} Ce'(u) = Ce(kz(u)) \\ Cf'(u) = Cf(kt(u)) \\ Cv'(u) = Cv(km(u)) \end{cases} \quad (18)$$

Step 8: Convert Ce′, Cf′ and Cv′to matrices Ew(n, b× 4), Fw(n, b× 4) and Vw(n, b×4), respectively. Decode Ew, Fw and Vw using a same rule $Rule_{dec}$ and get three binary matrices Ev, Fv and Vv.

Step 9: Finally, recover data that is the encrypted one

Decryption is the opposite of encryption in that it is a reverse operation. In order to decode the encrypted data, the receivers must get secret keys from the sender. Following is a step-by-step breakdown of the decryption procedure.

Step 1: Having $Rule_{dec}$, Components of the ciphered data are encoded. Three matrices were given as a input in total. Ew, Fw and Vw, as well as their vectorization into three Ce′, Cf′ and Cv′.

Step 2: Ce′, Cf′ and Cv′are the vectors. For obtaining the non- vectors Ce, Cf and Cv, step 8 of the encryption method is reversed as a result:

$$\begin{cases} Ce(u) = Ce'(kz(u)) \\ Cf(u) = Cf'(kt(u)) \\ Cm(u) = Cm'(km(u)) \end{cases} \quad (19)$$

Where, kz, kt and km It is addressed in the encryption procedure in stages 6 and 7.

Step 3: Convert the three vectors Ce, Cf and Cv in the form of the matrices Oe', Of'andOv'.

Step 4: According to XOR operation the invert process was done then the encryption algorithm as follows:Oe =Oe' XOR N_{jw}, Of = Of' XOR N_{je} and Ov = Ov' XOR N_{jw}, where N_{jw} utilizing the key sequence J, as described in encryption step 5, yields the desired result.

Step 5: Oe, Of and Ov are the three matrices for the sequence. Using the rule, the decoding was done (Oe, Of, and Ov)to extract the E, F, and V components of the data.

Step 6: Finally, the original data was recovered.

Before the data can post to the server it can be splitted to improve the second level of the security, for that the MHA was used.

Hellman Fibonacci sequence D_b can be defined as follows.

$$D_b = \begin{cases} 0 \; b = 0 \\ 1 \; b = 1 \\ D_{b-1} + D_{b-2} \; otherwise \end{cases} \quad (20)$$

Equation (20) is used to construct the Fibonacci sequence, which consists of the integers. Using any four consecutive terms of the Fibonacci numbers, a 2 x 2 matrix may be created that can be used to scramble data. The definition of a generalized Fibonacci mask is as follows:

$$\begin{bmatrix} \dot{z} \\ \dot{t} \end{bmatrix} = \begin{bmatrix} d_u & d_{u+1} \\ d_{u+2} & d_{u+3} \end{bmatrix} \begin{bmatrix} z \\ t \end{bmatrix} \text{ mod (b)} \quad (21)$$

Where z, t, $\dot{z}$, $\dot{t} \in$ 0,1,2,3,4, … ,b−1 , du is the u^{th} term of the Fibonacci series, and b is the overall data size z, A crammed dataset has a new coordinate for the original data, which is t. Scan the whole data horizontally and vertically so that the data may be scrambled into new information.

Y orthogonal matrix of size n×n, C orthogonal matrix of size b×b and A diagonal matrix of size n×b are all singular value decomposition matrices for any given matrix L ∈ $E^{n \times b}$ where u≠h.

$$L_{nb} = Y_{nn} A_{nb} C_{bb}^{R} \quad (22)$$

Where $Y^T Y=I$, $C^T C=I$ and $a_{11} \geq a_{22} \geq \cdots a_{oo} \geq 0$, where o = min {n, b}. The columns of **Y** are orthonormal eigenvectors of LL^T, the columns of **C** are represented as a orthonormal eigenvectors of $L^T L$, where A is a diagonal matrix that contains the square root of Eigenvalues from Y or C in decreasing order. This is what matrix N would look like if it was 5 by 3.

$$N = Y \times A \times C^T \tag{23}$$

$$\begin{bmatrix} n_{11} & n_{12} & n_{13} \\ n_{21} & n_{22} & n_{23} \\ n_{31} & n_{32} & n_{33} \\ n_{41} & n_{42} & n_{43} \\ n_{51} & n_{52} & n_{53} \end{bmatrix} = \begin{bmatrix} y_{11} & y_{12} & y_{13} & y_{14} & y_{15} \\ y_{21} & y_{22} & y_{23} & y_{24} & y_{25} \\ y_{31} & y_{32} & y_{33} & y_{34} & y_{35} \\ y_{41} & y_{42} & y_{43} & y_{44} & y_{45} \\ y_{51} & y_{52} & y_{53} & y_{54} & y_{55} \end{bmatrix} \begin{bmatrix} a_{11} & 0 & 0 \\ 0 & a_{22} & 0 \\ 0 & 0 & a_{33} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}^T \tag{24}$$

The following are the measures that will be taken:

1) First, need to take a look at the data in the form of a matrix Z.

2) Using Hellman Fibonacci Transform, randomize the elements of matrix Z to get matrix V.

3) Apply the SVD transformation to the matrix V to create three new matrices.

$$[Y, A, C] = SVD(V) \tag{25}$$

4) L and O are Diagonal matrices that contain only integer and fractional parts of A values, respectively.

$$A = L + O \tag{26}$$

5) The receiver already has the same keys (big and small) that may be utilized for the data reversing operation.

C. Equations

The attack success rate is done in a Scyther. Here $\alpha < \beta + \gamma$, the probability decreases exponentially with an increasing R_x . In this analysis, the hash power (G) will dominate the mining race in R_x time. When predicting how many blocks will be mined over the course of a certain period of time, the Poisson distribution may be used.

A double-spending assault happens when the opponent makes a second payment that is more than the previous payment. O_{sa} may be calculated by adding up all possible double spending attacks.

$$O_{so} = \sum_{J_1=0}^{+\infty}[O_1(J_1, R_x) \sum_{J_2=J_1+1}^{+\infty} O_2(J_2, R_x)], \tag{27}$$

Where, $O_1(J_1, R_x)$ and $O_2(J_2, R_x)$ are defined in Equation (27) and Equation (27), respectively.

It is the payee of the double-payment assault who has got the first payment and is trying to get a second one from Arbitrator. It is the payer's goal to defeat the payee in arbitration so that the arbitrator does not make a second payment to the latter. Hash power is used to create Non-Payment Proof in the double-payment attack βG and PaymentChallenge is generated with hash power $(\alpha + \gamma)G = (1 - \beta)G$ By the similar analysis as O_{sa}, the probability of double-payment attack O_{so} is given by.

$$O_{so} = \sum_{J_1=0}^{+\infty}[O_1(J_1, R_x) \sum_{J_2=J_1}^{+\infty}[O_2(J_2, R_x)], \tag{28}$$

Where, $O_1(J_1, R_x)$ and $O_2(J_2, R_x)$ are dened in Equation (27) and Equation (28), respectively. If Non-Payment Proof and Payment Challenge have the same length, then Arbitrator decides in favor of the payee, and there is no other difference in the computation of O_{so} and O_{sa}. With rising R_x, O_{so} decreases exponentially. It is therefore possible to minimize the O_{so} double-payment chance significantly by raising the parameter R_x by using the proposed procedure.

V. Performance Analysis

Mobile payment transactions between smartphones and payment terminals will be made more secure thanks to a new protocol were introduced in this paper. It provides an extra layer of cryptographic protection to address payment security flaws. It safeguards: the integrity and confidentiality of banking information, as well as the mutual authentication and non-repudiation of those data; and the validity of those data that are not. Using the Scyther tool, the suggested methodology was tested.

Humans still have a hard time proving that a security protocol is correct and safe. So, Scyther tool was used, which has been used in both research and teaching environments, to check the suggested protocol. Scyther enables security protocols to be analyzed in a formal manner by detecting possible attacks and weaknesses. When compared to other security verification methods, the results from Scyther's researchers are impressive. Unrestricted sessions and assured end-of-life are the hallmarks of Scyther's protocol analysis service. It generates a graph describing an assault if it discovers one that matches a certain claim. Protocols in the Scyther tool are implemented using the Security Protocol Description Language (SPDL). All Scyther statements that no attacks have been detected on Banking Data are supported by the protocol, as shown in Fig. 3. Definitions based on official sources are provided below.

The proposed Logical Diffie Hellman algorithm may be shown to be successful by comparing it to the [20] in order to demonstrate its efficacy.

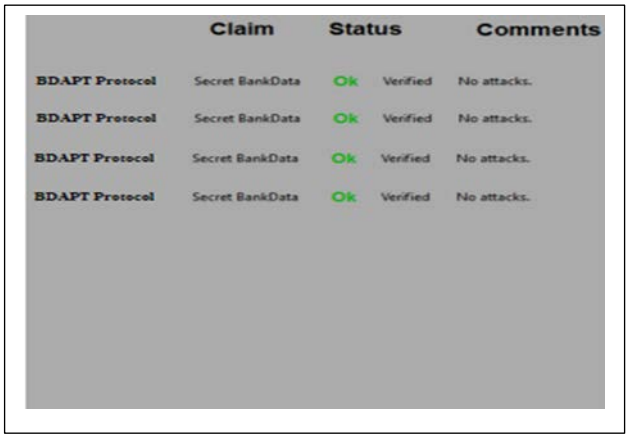

Fig. 3. Protocol Efficiency Analysis.

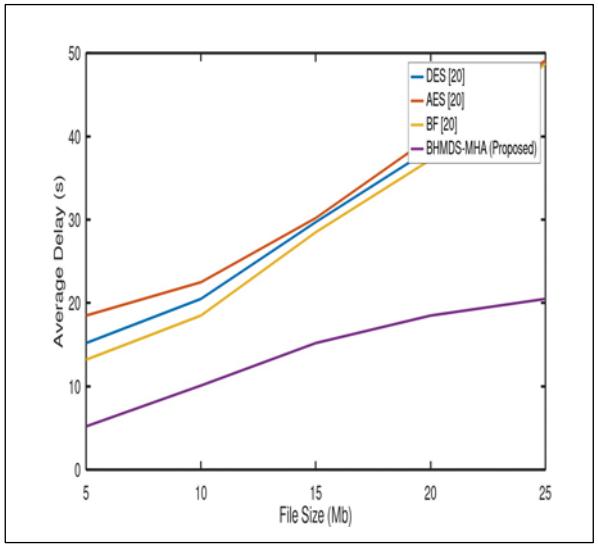

Fig. 4. File Size Vs. Average Delay.

Errors are more likely to occur during a switchover if the information used is inaccurate, imprecise, or confusing. Processing, queuing, transmission, and propagation delays are the four most common types of delays in packet switched networks. The quality and timeliness of a product are threatened by delays in communication (Transmission). Some of the probable repercussions include longer wait times, delays in discharge and poor decision-making. The delay ratio is shown in Fig. 4. The suggested method has a much lower delay ratio (20.5) than earlier procedures.

Fig. 5 and 6 depicts the time it takes to encrypt and decode a message in milliseconds, and the performance of the method is calculated. The suggested approach is compared to existing algorithms as BF, DES, and AES in terms of time differences.

It is shown in Fig. 7 that the proposed technique consumes less energy than other well-known encryption algorithms such as BF and DES.

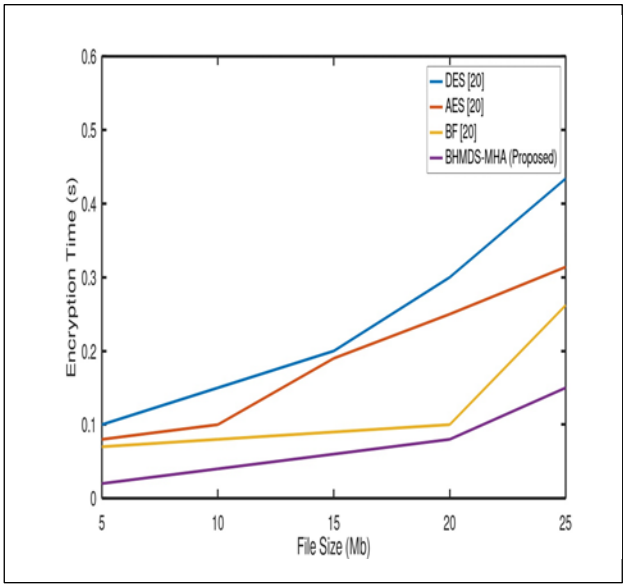

Fig. 5. File Size Vs. Encryption Time.

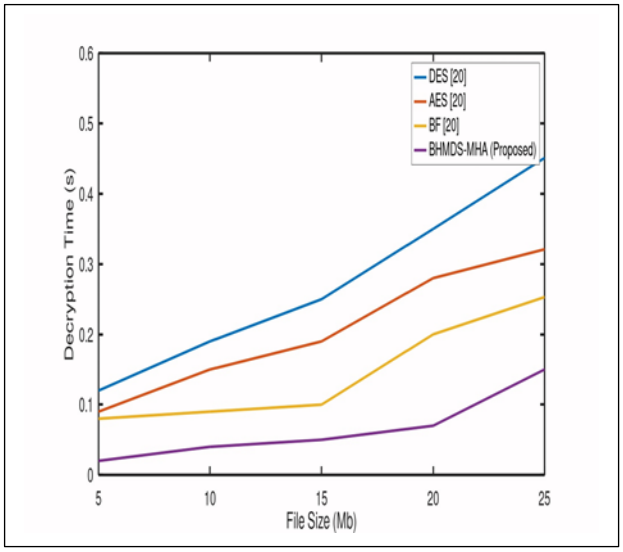

Fig. 6. File Size Vs. Decryption Time.

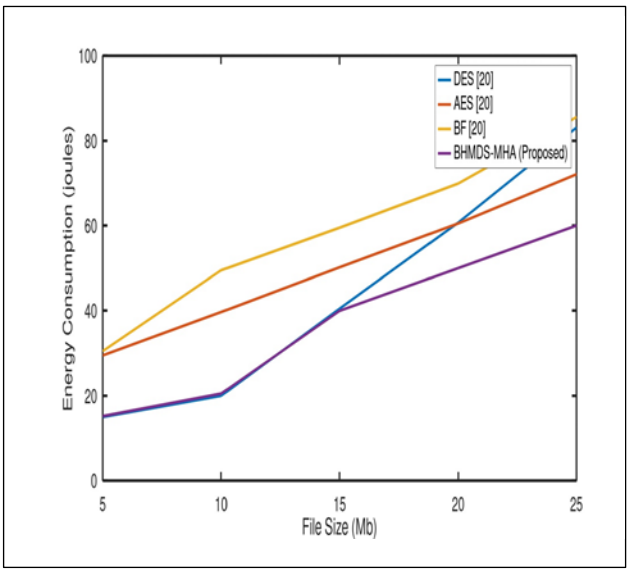

Fig. 7. File Size Vs. Energy Consumption.

Fig. 8 compares the proposed method to existing ones like DES BF and AES and shows how much faster it is in terms of throughput (in kbps). Fig. 5 to 8 show how the suggested method is tested. They look at things like average delay, throughput, encryption time, energy consumption, and decryption time to see how well the method works. BF, DES, and AES are used to figure out BHMD5-MHA, which takes into account things like average delay and throughput. BF is a competitor in near vicinity that ensures the privacy and integrity of user data. Although it seeks to reduce key complexity, its security is sometimes unreliable. BHMD5-MHA improves security for secure and successful financial transactions while also reducing the likelihood of fraud.

A secured mobile payment system was created here in this study by first employing the BHMD5-MHA method, which saves encrypted blocks of data and then links them together. An evaluation of the proposed mechanism's level of security is carried out. The proposed BHMD5-MHA algorithm is

compared against existing methods as DES [18], RSA [19], and AES [21].

Various encryption algorithms are compared and contrasted in Fig. 9 to show how secure they are in comparison. The existing DES, RSA, AES methodologies are compared with a new one called BHMD5-MHA. The security level is 82% TDH, 78% DES, 77% RSA, and 73% AES for files less than 20 MB. Security is also evaluated for file sizes as large as 40 MB and as little as 1 MB. When compared to various encryption algorithms, the graph shows that the BHMD5-MHA is the most secure. According to the data, the proposed technique beats other currently used procedures when it comes to ensuring transaction security.

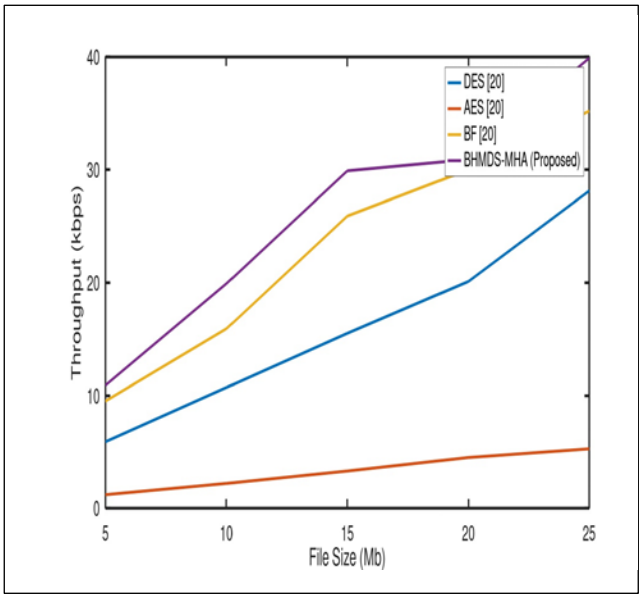

Fig. 8. File Size Vs. Throughput.

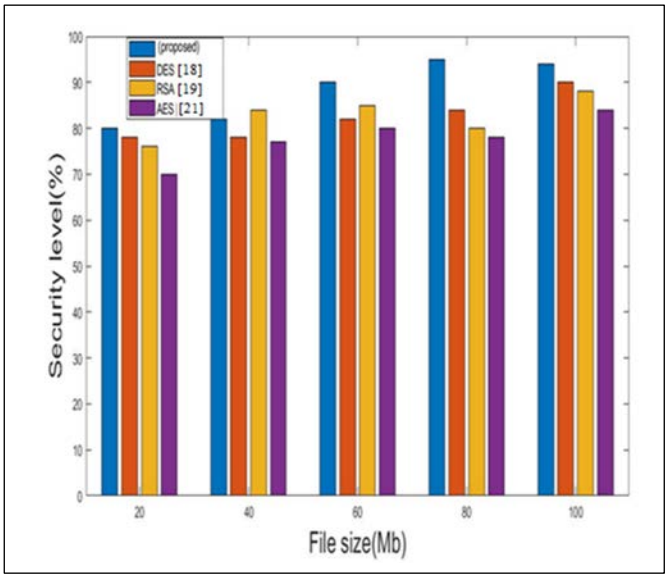

Fig. 9. File Size Vs. Security Level.

VI. Discussion

Based on the outcome that was achieved, our protocol meets the transaction security requirements listed below: 1) Symmetric encryption and the secret are used to guarantee the authenticity of the parties involved. 2) Transaction privacy is ensured by encryption, 3) Transaction integrity is ensured by the suggested protocol, and 4) Non-repudiation of transactions is also ensured in that the merchant is able to provide a non-repudiable evidence to prove to other parties that the client has originated the message. The encryption ensures that either the client or the merchant has originated the message and authenticates the client. Because the keys that are communicated between parties in our proposed protocol need to be updated on a regular basis or in response to specific requests, another worry that emerges relates to the distribution of keys. In the normal course of events, every time a new key is released, even if it is done so in an encrypted form, it is still feasible for an adversary to obtain it. Because it is possible to enlist the information in a coded form, our protocol does not need the customer to provide their card information. A participant in any transaction should not place their faith in other parties until those other parties can demonstrate that they can be trusted. Because the issuer is the one who provides the client with a credit card, our protocol specifies the trust connection that exists between the client and the issuer rather than the trust relationship that exists between the client and the payment gateway.

VII. Conclusion

The novel protocol and cryptographic techniques have been clearly addressed in this study. The Scyther compliance with standards and guidelines is used to demonstrate the model. An analogy of the key length is provided, as well as an analytical look into cryptography approaches for real-time use Due to the magnitude of the BHMD5's key, it has been demonstrated to be the most secure. As a second degree of protection, the MHA scrambles the original data. According to other symmetric key algorithms, this approach is the most secure (96 percent) and uses the least processing time. According to the findings, the proposed method could help in deal with the growing security issues that come with online transactions.

Acknowledgment

This research was supported by the Department of Computer Science and Engineering, SRM Institute of Science and Technology, Kattankulathur, Chennai, Tamilnadu, India.

References

[1] F. Buccafurri and G. Lax, "Implementing disposable credit card numbers by mobile phones," Electronic Commerce Research, vol. 11, pp. 271-296, 2011.

[2] A. Braeken, "Public key versus symmetric key cryptography in client–server authentication protocols," International Journal of Information Security, vol. 21, pp. 103-114, 2022.

[3] S. Bojjagani, V. Sastry, C.-M. Chen, S. Kumari, and M. K. Khan, "Systematic survey of mobile payments, protocols, and security infrastructure," Journal of Ambient Intelligence and Humanized Computing, pp. 1-46, 2021.

[4] E. Erdin, S. Mercan, and K. Akkaya, "An evaluation of cryptocurrency payment channel networks and their privacy implications," arXiv preprint arXiv:2102.02659, 2021.

[5] H. Li, T. Wang, Z. Qiao, B. Yang, Y. Gong, J. Wang, et al., "Blockchain-based searchable encryption with efficient result verification and fair payment," Journal of Information Security and Applications, vol. 58, p. 102791, 2021.

[6] J. Zhang, Y. Ye, W. Wu, and X. Luo, "Boros: Secure and Efficient Off-Blockchain Transactions via Payment Channel Hub," IEEE Transactions on Dependable and Secure Computing, 2021.

[7] K.-H. Yeh, C. Su, J.-L. Hou, W. Chiu, and C.-M. Chen, "A robust mobile payment scheme with smart contract-based transaction repository," IEEE Access, vol. 6, pp. 59394-59404, 2018.

[8] Y. Chen, W. Xu, L. Peng, and H. Zhang, "Light-weight and privacy-preserving authentication protocol for mobile payments in the context of IoT," IEEE Access, vol. 7, pp. 15210-15221, 2019.

[9] M. Ramachandran and V. Chang, "Towards performance evaluation of cloud service providers for cloud data security," International Journal of Information Management, vol. 36, pp. 618-625, 2016.

[10] F. Corradini, L. Mostarda, and E. Scala, "ZeroMT: Multi-transfer Protocol for Enabling Privacy in Off-Chain Payments," in International Conference on Advanced Information Networking and Applications, pp. 611-623, 2022.

[11] N. El Madhoun and G. Pujolle, "Security enhancements in emv protocol for nfc mobile payment," in IEEE Trustcom/BigDataSE/ISPA, pp. 1889-1895, 2016.

[12] S. Abughazalah, K. Markantonakis, and K. Mayes, "Secure mobile payment on NFC-enabled mobile phones formally analysed using CasperFDR," in 13th International Conference on Trust, Security and Privacy in Computing and Communications, pp. 422-431, 2014.

[13] Y.-Y. Chen, M.-L. Tsai, and F.-J. Chang, "The design of secure mobile coupon mechanism with the implementation for NFC smartphones," Computers & Electrical Engineering, vol. 59, pp. 204-217, 2017.

[14] S.-W. Chen and R. Tso, "NFC-based mobile payment protocol with user anonymity," in 11th Asia Joint Conference on Information Security (AsiaJCIS), pp. 24-30, 2016.

[15] N. El Madhoun, F. Guenane, and G. Pujolle, "An online security protocol for NFC payment: Formally analyzed by the scyther tool," in Second International Conference on Mobile and Secure Services (MobiSecServ), pp. 1-7, 2016.

[16] N. El Madhoun, F. Guenane, and G. Pujolle, "A cloud-based secure authentication protocol for contactless-nfc payment," in 4th International Conference on Cloud Networking (CloudNet), pp. 328-330, 2015.

[17] J. N. Luo, M. H. Yang, and S.-Y. Huang, "An Unlinkable Anonymous Payment Scheme based on near field communication," Computers & Electrical Engineering, vol. 49, pp. 198-206, 2016.

[18] R. Shivhare, R. Shrivastava, and C. Gupta, "An Enhanced Image Encryption Technique using DES Algorithm with Random Image overlapping and Random key Generation," in International Conference on Advanced Computation and Telecommunication (ICACAT), pp. 1-9, 2018.

[19] V. Rao, N. Sandeep, A. R. Rao, and N. Niharika, "FPGA Implementation of Digital Data using RSA Algorithm," Journal of Innovation in Electronics and Communication Engineering, vol. 9, pp. 34-37, 2019.

[20] A. H. Be and R. Balasubramanian, "Encryption Algorithm for High-Speed Key Transmission Technique," Advances in Dynamical Systems and Applications, vol. 16, pp. 1557-1267, 2021.

[21] X. Dong, D. A. Randolph, and S. K. Rajanna, "Enabling privacy-preserving record linkage systems using asymmetric key cryptography," in AMIA Annual Symposium Proceedings, pp. 380, 2019.

[22] A.Saranya, R.Naresh "Cloud-Based Efficient Authentication for Mobile Payments using Key Distribution Method", Journal of Ambient Intelligence and Humanized Computing, 2021. [Online]. Availble: http://dx.doi.org/10.1007/s12652-020-02765-7.

[23] A.Saranya, R.Naresh "Efficient mobile security for E-healthcare application in cloud for secure payment using key distribution", Neural Processing Letters, 2021. [Online]. Availble: http://dx.doi.org/10.1007/s11063-021-10482-1.

[24] R.Naresh, P.Vijayakumar, L. Jegatha Deborah, R. Sivakumar, "A Novel Trust Model for Secure Group Communication in Distributed Computing", Special Issue for Security and Privacy in Cloud Computing, Journal of Organizational and End User Computing, vol. 32, no. 3, 2020.

[25] R.Naresh, M.Sayeekumar, G.M.Karthick, P.Supraja, "Attribute-based hierarchical file encryption for efficient retrieval of files by DV index tree from Cloud using crossover genetic algorithm", Soft Computing, Springer, vol.23, no. 8, pp. 2561-2574, 2019.

[26] R.Naresh, AyonGupta, Sanghamitra, "Malicious Url Detection System Using Combined Svm And Logistic Regression Model", International Journal of Advanced Research in Engineering and Technology, vol. 10, no. 4, pp. 63-73, 2020.

[27] Gautam Srivastava, C.N.S. Vinoth Kumar, V Kavitha, N Parthiban, Revathi Venkataraman, "Two-Stage Data Encryption using Chaotic Neural Networks", Journal of Intelligent and Fuzzy Systems, vol. 38, no. 3, pp. 2561-2568, 2020.

[28] R. Mugesh, "A Survey on Security Risks in Internet of Things (IoT) Environment," Journal of Computational Science and Intelligent Technologies, vol. 1, no. 2 pp. 01-08, 2020. [Online]. Available: https://doi.org/ 10.53409/mnaa.jcsit20201201.

[29] R. Sathiyasheelan, "A Survey on Cloud Computing for Information Storing," Journal of Computational Science and Intelligent Technologies, vol. 1, no. 2 pp. 09-14, 2020. [Online]. Available: https://doi.org/ 10.53409/mnaa.jcsit20201202.

[30] A.N. Suresh, "A Hybrid Genetic-Neuro Algorithm for Cloud Intrusion Detection System," Journal of Computational Science and Intelligent Technologies, vol. 1, no. 2 pp. 15-25, 2020. [Online]. Available: https://doi.org/ 10.53409/mnaa.jcsit20201203.

[31] C.N.S.Vinoth Kumar, and A.Suhasini, "Secured Three-Tier Architecture for Wireless Sensor Networks Using Chaotic Neural Networks", Advances in Intelligent Systems and Computing, vol. 507, no. 13, pp. No. 129-136, 2017. [Online]. Available: https://doi.org/10.1007/978-981-10-2471-9_13.

[32] C.N.S.Vinoth Kumar, and A.Suhasini, "Improved secure three-tier architecture for WSN using hop-field chaotic neural network with two stage encryption," in International Conference on Computer, Electrical & Communication Engineering 2016. [Online]. Available: https://doi.org/*10.1109/ICCECE.2016.8009540*.