Encryption and Tokenization-Based System for Credit Card Information Security

Article *in* International Journal of Cyber-Security and Digital Forensics · August 2018

DOI: 10.17781/P002462

3 authors:

Gabriel Iwasokun
Federal University of Technology
63 PUBLICATIONS **389** CITATIONS
SEE PROFILE

Taiwo Gabriel Omomule
Adekunle Ajasin University
23 PUBLICATIONS **128** CITATIONS
SEE PROFILE

Raphael Olufemi Akinyede
Federal University of Technology
31 PUBLICATIONS **85** CITATIONS
SEE PROFILE

Encryption and Tokenization-Based System for Credit Card Information Security

Gabriel Babatunde Iwasokun[1], Taiwo Gabriel Omomule[2], Raphael Olufemi Akinyede[3]

[1]Department of Software Engineering, Federal University of Technology, Akure, Nigeria
[2]Department of Computer Sciences, Adekunle Ajasin University, Akugba-Akoko, Nigeria
[3]Department of Information System, Federal University of Technology, Akure, Nigeria
[1]gbiwasokun@futa.edu.ng, [2]taiwo.omomule@aaua.edu.ng , [3]roakinyede@futa.edu.ng

ABSTRACT

Existing methods for promoting credit card information security have exhibited confidentiality, privacy and integrity failures. In most cases, sensitive and classified data or information is presented in unencrypted formats to remote machines resulting in unauthorized access and disclosure. Based on these findings, this paper presents an RSA encryption and tokenization-based system for credit card information security. The proposed system is composed of the merchant and tokenization modules as well as a token vault. The merchant and the tokenization modules send credit card information and generate the transaction validation token respectively while the token vault is a database with restricted and controlled access hosted on a cloud storage engine. The implementation of the system was carried out on Pentium IV with 2.0 GHZ Duo Core Processor and 2 GB of RAM on Microsoft Windows 7 Operating System. APACHE server and HTML (Sublime) with CSS JavaScript served as the frontend while MySQL database from WAMP server and PHP joined as the backend on Mozilla Firefox browser. Analysis of the results of implementation with Master, Verve and Visa cards showed that the system delivered very high usability, adaptability and favorable experience for users. Analysis also showed the relative advantages and superiority of the system in credit card security, key size, mobile alert and tokenization over some other systems.

KEYWORDS

RSA, financial crime, credit card, tokenization, encryption.

1 INTRODUCTION

Consequent to the rapid growth in Information Technology (IT) industry in recent years, the number of Internet users has witnessed explosive growth. Internet has offered great opportunity for improved communication and experience to the global citizens. It has also provided new drives for credit-card-based businesses and marketing by financial institutions towards reaching out to new markets and creating opportunities for economic growth. Credit card payment processing (conceptualized in Figure 1) involves the submission of relevant information such as card number, payment fee, expiry date and so on to the merchant for verification and submission to the Payment Gateway (PG). The PG passes the received information to the processor which then establishes a linkage with the customer's bank for a decision on whether to push or drop the request. Money is not immediately transferred to the merchant's bank, but instead, a token of settlement, which is a merchant's electronic payment information for a certified transaction, is delivered [1]. The proliferation of Internet-based businesses has before time been largely attributed to the rising confidence levels and trust among participants and the extent to which information confidentiality can be maintained. However, in view of the emerging range of fraud, theft, disruption and denial of service attacks on online transactions, stakeholders have expressed great concern, doubt and loss of confidence on credit card information transmission, privacy, integrity and security [2].

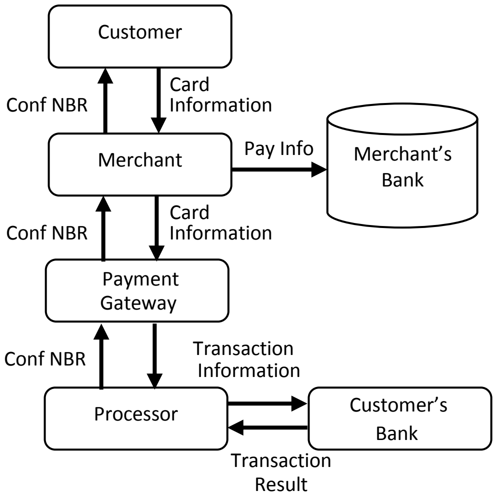


Figure 1: Payment process model for Credit Card

Unauthorized access, information theft and other breaches have exposed the security weaknesses in the existing and traditional Internet-based credit card systems with the attendant effect been loss of several millions of dollars annually [3-4]. A credit card fraud is committed when an individual inadvertently and callously use the information on the card for deception, misrepresentation and other selfish and personal reasons. It may be committed through lost/stolen card, account takeover, Cardholder-Not-Present (CNP), magnetic stripe erasure, card counterfeiting and skimming. Commonly used techniques in Internet credit card fraud include site cloning, false merchant sites and unlawful credit card generation. Merchant related credit card frauds may be in form of collusion (merchant and/or employees conspire to commit fraud using customers' accounts details and/or personal information) or triangulation (fraudsters use customer's personal information and valid credit card details to place order for goods from a recognized site). The risks associated with credit card fraud often imparted jointly on the merchant and the cardholder, who may be unaware of the attack. While cardholders are faced with the daunting challenge of getting a fraud-related charge reversed, merchants are confronted with sales lost arising from pay-back fees as well as threat of account closure or suspension [5].

2 CREDIT CARD PROCESSING

A number of methods have been suggested for securing online-based credit card transactions. Secure Socket layer (SSL)-based encryption has been used for the prevention of eavesdropping during the transmission process. This method relies on asymmetric key encryption for customer-merchant communication and promotes digital certificate-based authentication of the identity of the merchant [6]. Secure Electronic Transaction (SET) has been proposed as a protocol for ensuring secured credit card payment over the Internet. It operates by establishing protocols for cardholder and merchant registration, purchase request, payment authorization and payment capture [1]. In contrast to SSL, SET prevents illegal use of credit card number through the enforcement of exclusive sharing of information on transaction order and payment information with the merchant and bank respectively (dual signature). SET is however noted for its complexity which often times results in some incidences of credit card insecurities over the Internet. Several other algorithms such as encryption algorithm and cryptography with significant advantages over SSL and SET had been formulated for the safety and protection of credit card information. Encryption algorithm is used to transfer card or transaction information into a form that makes it impossible to read without the appropriate knowledge (a key). It ensures privacy by keeping information hidden from intruders and impostors. The transformation of encrypted data back into an intelligible format requires decryption. Same key may be required for encryption and decryption in some cases while they may differ in others. Cryptography is classified into secret key and public key. Secret-key cryptography is also known as symmetric cryptography and works with same key for encryption and decryption (as shown in Figure 2) [7] and the most popular ones are Data Encryption Standard (DES), Advanced Encryption Standard (AES), Triple DES, Blow Fish, Two Fish, Three Fish among others [8-20].

These techniques require a priori knowledge of the message and its key by the sender and receiver for encryption and decryption respectively and their main challenge is the complexity that trails the enforcement of the exclusive agreement on the secret key between the sender and receiver. Furthermore, the key generation, transmission and storage often require established and complex management issues [21]. Public-key cryptography is established for encryption and digital signatures with each person allotted public and private keys as shown in Figure 3 [7]. While public key is published and associated with users in a trusted (authenticated) manner, the private key is made secret based on non sharing of information between the sender and receiver thereby promoting privacy (encryption) and authentication (digital signatures) [7, 22-23]. Common public key algorithms include Elliptic Curve Cryptography (ECC) and RSA algorithms [24-25].

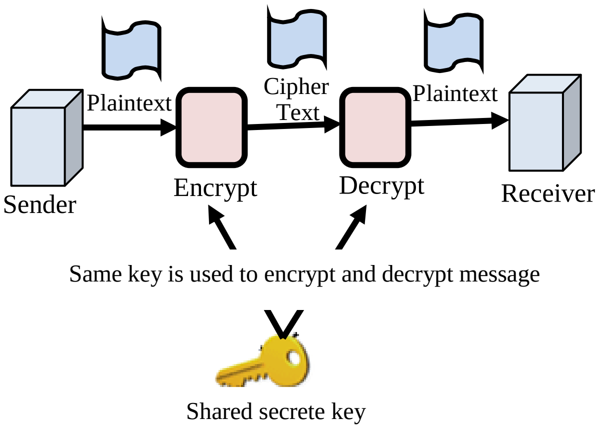


Figure 2: Private key cryptography

Encryption strength is directly tied to key size as doubling the key length delivers an exponential increase in strength, although it does impair performance. The weaknesses of the encryption and cryptography algorithms further include large space requirement, susceptible to cipher text and cycle attacks, common modulus, cracking, low exponent and complexity as well as financial, bandwidth, power and delay overheads.

3 RELATED WORKS

A modeling and classification approach for credit card fraud detection in electronic payment services is presented in [17]. The problem of fraud in online payment services was addressed using a whistle blowing strategy. A model of the history of legitimate transactions by entities such as the buyer, seller, card and cardholder was developed along a Support Vector Machine (SVM)-based method for the classification of the transactions as fraud or legitimate. Though the proposed method is applicable in detection of credit card frauds based on historical analysis of financial data, it offers no real-time support and could not analyze the behavior of the entities.

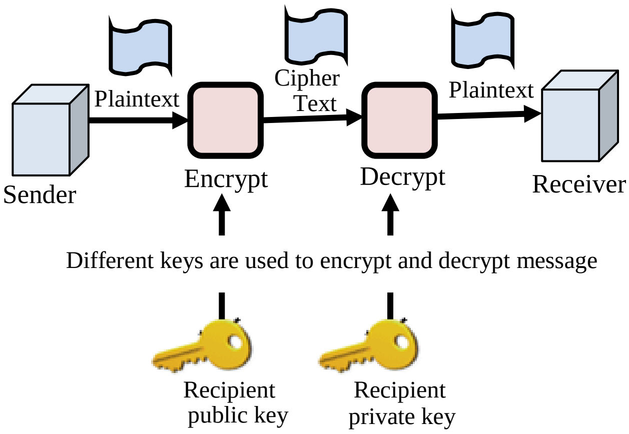


Figure 3: Public key cryptography

The authors in [26] proposed a fuzzy clustering and neural network–based credit card fraud detection system. The research analyzed the spending pattern of credit card users and classified transactions as fraudulent or genuine based on deviation from usage patterns. A fraud detection model is abstractly represented as a 5-tuple {C, P, SC, ϕ_{Uth}, ϕ_{Lth}}, where C = {C_1, C_2, …, C_n} is the set of cards on which the detection is performed and P = {P(C_1), P(C_2), …, P(C_n)} is the set of card holders profiles. The profile is a set of patterns on transaction amount, items purchased, time of transaction and so on. SC is the suspicion score for measuring the extent of deviation from normal profile, ϕ_{Uth} `is the upper threshold value for $0 \leq \phi_{Uth} \leq 1$ and ϕ_{Lth} is the lower threshold value for $0 \leq \phi_{Lth} \leq 1$ and $\phi_{Lth} \leq \phi_{Uth}$. The hybridization of Fuzzy C-Means Clustering and Neural Network for credit card fraud detection is presented in Figure 4 showing a multi-layered approach. The research established a model for the investigation of the neurological and behavior pattern in a credit card transaction but lacks capability for implementing attributes such as location of transaction and time gap between transactions which are very relevant for improved security rules.

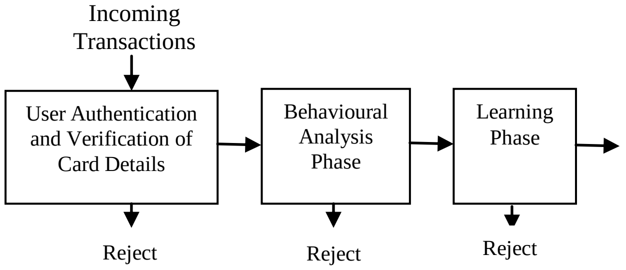

Incoming Transactions
User Authentication and Verification of Card Details
Behavioural Analysis Phase
Learning Phase
Reject
Reject
Reject

Figure 4: Prototype Structure for credit card fraud detection

In [15], a cloud-based system for local data integrity and privacy protection is presented. The system uses BlowFish encryption algorithm to encrypt local clients-domiciled data so that a perfect data security and privacy protection mode is established without infringing on the experience and efficiency of use. A modified encryption algorithm was developed for addressing some of the existing problems of data security, but its application is limited to local encryption of data. The authors in [19] developed a voucher/certificate and bilinear pairing cryptography-based system for data, information and communication safety. Bilinear pairing and symmetry key were used for encryption of the communicated messages and a problem-free transfer of data on the cloud respectively. Trusted third party was also used to establish a security domain on the cloud while ID based blind signature was adopted for the attainment of required security level. Though the system afford users the opportunity of self established level of security, it does not support concurrent access control on personal user's security activities. In [10], a flask architecture-based model for personalized security in cloud-based e-banking is introduced. The model provided a role-based reconfigurable access control strategy that ensures safe and secured e-payment system based on user-defined policies. The model demonstrated ability for personalized security in cloud environment but improper configuration for user policies may constitutes stumbling blocks to its implementation. A DES-based system for encryption and decryption in network data transmission is presented in [9]. The system is capable of analyzing data security risk and requirements as well as deploying security functions and process through encryption. Cipher Block Chaining (CBC)-based DES was used to encrypt information in transition on the network with each block of Electronic Code Book (ECB) encrypted cipher text XORed with the next plain text block for encryption, thus ensuring dependency between current and previous blocks. The system promoted data security in cloud computing environment but error transmission across blocks is still a possibility.

In [20], an encryption-based model for secured cloud storage access and query is presented. The system focuses on query service, privacy protection, authentication management, data storage as well as integrity protection in the cloud environment. With this model, only cloud service-oriented storage and query users are allowed to interact through cloud storage and the management servers. The server ensures privacy based on trusted third party user authentication and information management. It is also in charge of key management and distribution, safe registration of data as well as storage of queries results, cipher text and hash table of keyword. The strength of the model lies in its support for data privacy based on symmetric encryption while its weakness is its susceptibility to security breach and conflict arising from key management between the cloud users. The authors in [14] proposed a data confidentiality and privacy execution model which utilizes the public and private clouds for non-sensitive and sensitive data respectively. The model supports application level partitioning and allows integrations of additional computing or storage resources to the private cloud from public clouds without compromising the confidentiality or privacy of data. The model is suitable for the enhancement of data security benefits but its avoidance of public clouds for sensitive data and computation is a major concern. In [16], a token-based model for securing outsourced data and arbitrary computations with lower latency on the cloud computing environment is presented. The model is based on tamper-proof hardware token and encryption by homomorphism and supports a resource constrained tamperproof hardware token in the setup pre-processing phase and performs symmetric cryptographic operations in parallel in the online phase. The model promotes data confidentiality, integrity and verifiability but lacks capabilities to protect data on transmission in the cloud. In [18], a system that is based on homomorhpic token with distributed verification of erasure-coded data as well as signature generation and matching is developed for addressing the problem of insecurity in cloud data storage. The system ensures redundancy of parity vectors and data dependability using error-correcting code in file distribution preparation. The system also guarantees the integrity of cloud-based data using two-way handshake and provides a schematic approach to resisting Byzantine failure, malicious data modification and server colluding attack. The weakness of the system is expressed in its failure to consider the security of data in transition. In [13], a number-less credit card Kerberos Cryptographic-based system is presented as a solution to the rising cases of credit card frauds. A secured electronic payment system that conceals credit card number and makes it exclusive to the holder was developed. Tokens were generated based on fee, client and merchant information. The system uses Kerberos authentication protocol to exchange tokens between buyers and merchants. Though the research contributed to securing credit card transactions, the system lacks capabilities for implementation in a distributed environment due to its low operational speed and complexity. A credit card security system for e-payment is presented in [8]. The research was premised on the need to curtail the rising trend of hacking of credit card details on the Internet as well as provide a mechanism for different commercial institutions to minimize all attendant problems of credit card transactions. The research methodology involves a questionnaire-based survey of banks as well as inferential and descriptive statistics-based analyses and interpretations. Though the authors claimed some success in credit card frauds reduction, the research only presented theoretical basis with no evidence of practical process.

In [4], a practical security enhancement scheme for one-time credit card payment is presented. A cryptographic hash function-based credit card number and transaction verification algorithm was formulated. The hash function was used to generate a one-time credit card number and a secret pin, S that is exclusive to the card holder and issuer. The transaction verification model considers four transactions T_0, T_1, T_2, T_3 ordered on the processing time and for cryptographic function χ, $T_1 = \chi(T_0||S)$, $T_2 = \chi(T_1||S)$, $T_3 = \chi(T_2||S)$. T_0 is assumed to have been verified by the card issuer most recently and for transactions T_1, T_2 and T_3 all in instant verification scenario, then the transactions arriving for verification have the same order as their transaction times and the card issuer is only required to compute the hash chain and verify sequentially. A security analysis model that is based on the probability that an attacker knows a single verifiable transaction is presented as follows:

$$Max\left[\frac{(|Q|+n)}{2^{|s|}},\frac{(|Q|+n)}{10^{|T|}}\right] \tag{1}$$

$|Q|$, $|S|$, $|T|$ denote the length of Q, S or T respectively and the length of Q is constrained by a queueing policy. The algorithm permits the verification of at most m transactions verification and the probability of success is presented as follows:

$$Max\left[\frac{m(|Q|+n)}{2^{|s|}},\frac{m(|Q|+n)}{10^{|T|}}\right] \tag{2}$$

The model was simulated based on the Poisson distribution process of the exponential distribution time between two payment transactions and the delayed verification and its effectiveness for one-time credit card number and transaction verifications was established. However, since the model is based on hashing function, it is susceptible to failure in cases of sophisticated attacks such as replay attacks. Furthermore, due to transmission error or network disconnection, a transaction may not be delivered to its merchant site or card issuer site and this may result in confirmation failure

4 PROPOSED SYSTEM

The proposed system addresses some of the limitations of the reviewed and reported works and is composed of the merchant and tokenization modules as well as the token vault as conceptualized in Figure 5.

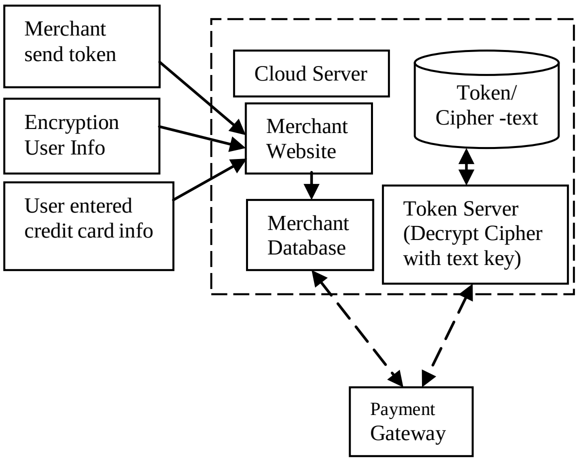


Figure 5: Architectural design of the proposed system

The merchant module provides the platform for a customer (a credit or debit card holder) to send credit card information while the tokenization module is used to generate transaction validation token and is controlled by the payment gateway. The token vault is a database with restricted and controlled access hosted on a cloud storage engine. As shown in Figure 6, a credit card number consists of the bank identification number (BIN), account number and the check digit which are used for identification.

4.1 Card Number Validation

The merchant validates the credit card number based on Luhn mathematical formula. Given that S = {A_1, A_2, ..., A_{n-1}, A_n} represents a sequence of credit card with *n* digits. The sum,

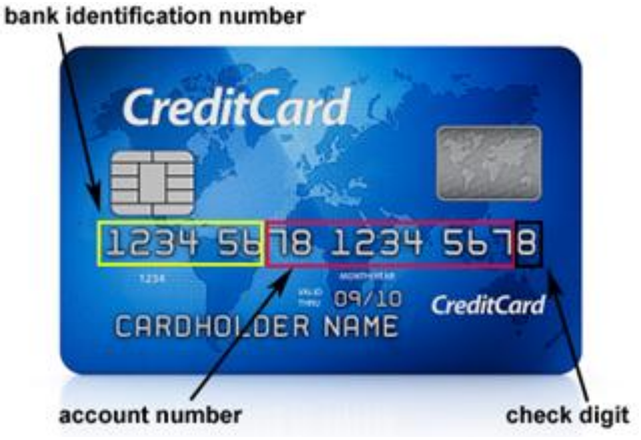


Figure 6: Credit Card Number

S_1 and S_2 of the digit of the odd and even products respectively, are computed as follows:

$$S_1 = \sum_{k=1}^{n-1} l_k \; ; \begin{cases} l_k \, , if \; l_k < 10 \\ \quad l_k = (l_k \backslash 10) + (l_k \, mod \, 10), \quad if \; l_k \geq 10 \end{cases} \qquad (3)$$

$$S_2 = \sum_{k=2i}^{n-1} (A_k), \quad \forall i = 1 \ldots n-1 \qquad (4)$$

$$Z = (S_1 + S_2) + A_n \qquad (5)$$

l_k and A_e are obtained by doubling the digits that occupy the kth odd and even positions respectively. The card number is validated if *Z mod 10 = 0*. The credit card number is encrypted based on a cryptographic scheme given as follows:

$$E(M) \dashrightarrow C \qquad (6)$$

$$D(C) \dashrightarrow M \qquad (7)$$

E, M, C and D represent Encryption, Message (Plaintext), Ciphertext and Decryption respectively. Rivest-Shamir-Adleman (RSA) encryption algorithm which is based on the combination of prime factorization, Euler's totient function, Euler's totient theorem and Extended Euclidean Algorithm (EEA) is used to compute the private key for decryption process. Prime Factorization is a fundamental theorem of arithmetic that establishes that any number greater than 1 can be written exactly one way as a product of prime numbers. For a prime number p, Euler's Totient Function ϕ is expressed as *$\phi(p)$ = p-1.* For primes p and q,

ϕ(p.q) = (p-1)(q-1). Euler's Totient Theorem states that $\phi(p.q) = (p-1)(q-1)$. The RSA algorithm involves three steps; namely key generation, encryption and decryption as presented in Figure 7.

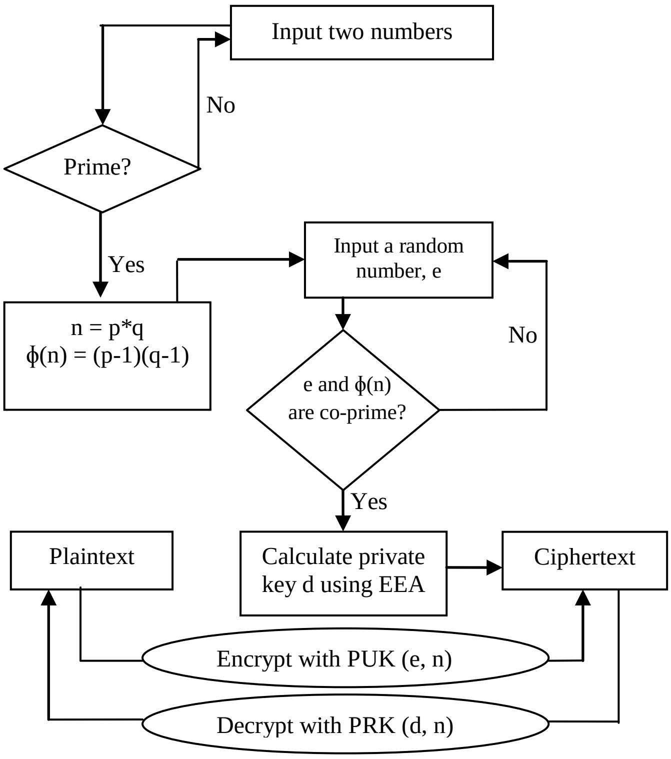


Figure 7: Flowchart of RSA Algorithm

The gateway sends the cipher-text to the tokenization server for decryption using RSA algorithm with Extended Euclidean Algorithm (EEA)-based private key. EEA computes a private key, *A* based on the matrix iterative scheme using *ϕ(n)* and a public key *e* as follows:

$$A = \begin{bmatrix} \phi(n) & \phi(n) \\ e & 1 \end{bmatrix} \quad (8)$$

4.2 Tokenization

A token is generated for the credit card number, and stored alongside the cipher text in the token vault (Token database). The tokenization module decrypts the cipher text and generates tokens from the real card information. During tokenization, a random number generator is used to generate a sequence of integers. For each credit card number starting from the left position to the 6th digit, the last four digits to the right is used as a placeholder for the tokenization of the real card number. Given that *i* = 1 to U, where *i* is a counter for numbers of transactions and U is the transaction upper bound, then:

for *i* = 1 (first transaction to generate a token)

$$T_1 = (x.A_k + y)\ mod\ A_k, \forall\, n,k = 1 \dots j, \forall\, x,y = 0 \dots 9 \quad (9)$$

for *i* = 2; second transaction

If the same credit card number is used, then generate a new token T_2 from token T_1, else generate token T_2 from the card number

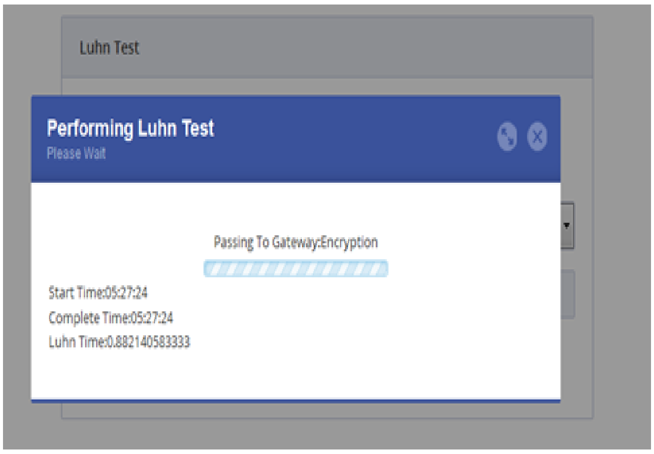
for *i* = U; upper-bound transaction

If the same credit card number is used, token T_U is generated from token T_{U-1}, else generate token T_U from the card number

A_k is a credit card number, T_i, i=1, 2, … U is the token generated, x and y are any random single digit number and j is the 6th digit. The notation *mod* A_k indicate that the expression in parentheses is divided by A_k, and then replaced with the remainder. The parameter A_k and the choice of x *and* y determine the characteristics of the random number generator.

5 IMPLEMENTATION

The implementation of the RSA encryption and tokenization-based platform for securing credit card information was carried out on Microsoft Windows 7 Operating System environment on Pentium IV with 2.0 GHZ Duo Core Processor and 2 GB of RAM. APACHE server and HTML (Sublime) with CSS, JavaScript served as the frontend while MySQL database from WAMP server and PHP were the backend on Mozilla Firefox browser. The web-based system is authentication-based and is divided into the user and the admin sessions. The user session takes the user through the home, registration, login, Luhn test, encryption, decryption, tokenization and confirmation interfaces. The home screen provides the system information while the registration interface is used for user's registration and setup. It leads to the login interface which is used to validate the authenticity of the user through supply of pre-registered username and password. The Luhn Test interface is used to investigate the validity of a Master, Verve or Visa Card number and as a first step of the RSA-based encryption as shown in Figure 8.



Luhn Test

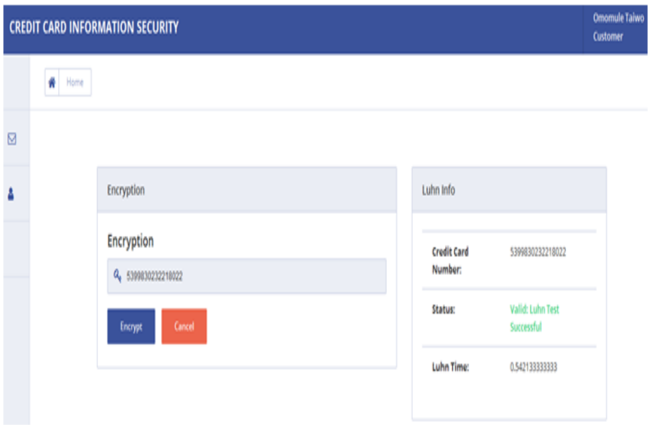
Performing Luhn Test
Please Wait

Passing To Gateway:Encryption

Start Time:05:27:24
Complete Time:05:27:24
Luhn Time:0.882140583333

Figure 8: Luhn Test

This test is terminated with the display of the Encryption interface that shows the credit card number being validated, Luhn time and status as well as success or failure report as shown in Figure 9. At a successful completion of the modal process for encryption, the decryption page is displayed with the obtained ciphertext as shown in Figure 10. The credit card number is not displayed because it is currently encrypted



CREDIT CARD INFORMATION SECURITY
Omomule Taiwo
Customer

Home

Encryption

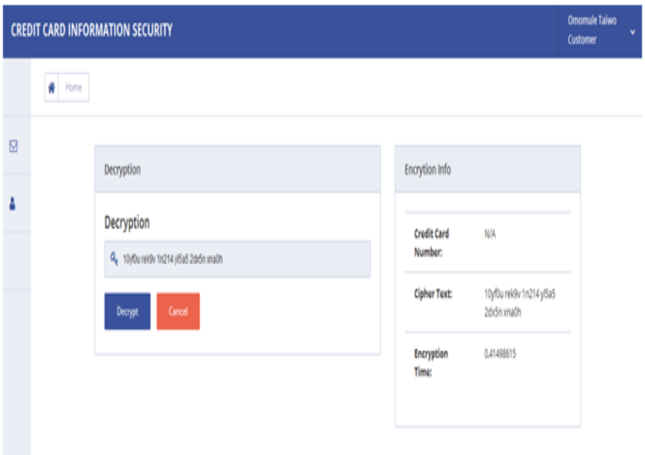
Encryption
5399830232218022

Luhn Info

Credit Card Number: 5399830232218022
Status: Valid: Luhn Test Successful
Luhn Time: 0.542133333333

Figure 9: The encryption interface

The aftermath of decryption is the display of the tokenization interface for a 6-digit token to be generated and sent to the user's pre-registered mobile phone number for acknowledgement and confirmation based on request screen shown in Figure 11. The system keep logs of users' information such as date and time of operation, token generated, token received and acknowledgement times as well as the new modulus and public keys generated for RSA algorithm-based operation on a Master, Verve or Visa credit card information.



CREDIT CARD INFORMATION SECURITY
Omomule Taiwo
Customer

Home

Decryption

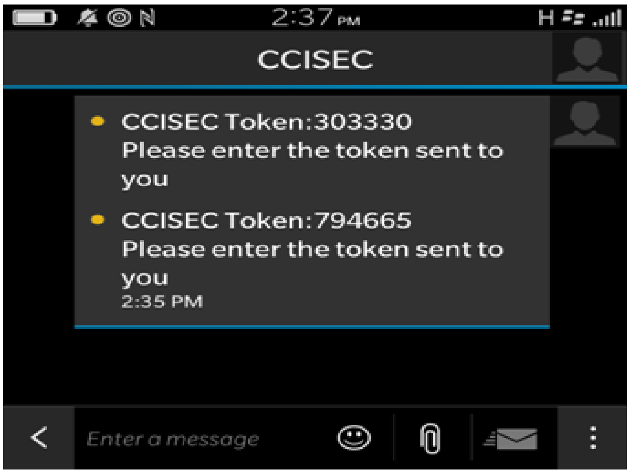
Decryption
10yf0u rek9v 1n214 yl5a5 2dx5n xna0h

Encrytion Info

Credit Card Number: N/A
Cipher Text: 10yf0u rek9v 1n214 yl5a5 2dx5n xna0h
Encryption Time: 0.41498615

Figure 10: Decryption interface

Figure 12 presents the average time taken to complete the encryption, decryption, tokenization and the confirmation processes for 3080 randomly selected users from financial and educational institutions and agencies that participated in the experimental analysis of the system. Table 1 presents the computation times (based on the prescribed algorithms) for Luhn test, encryption, decryption and tokenization in microseconds for Master, Verve and Visa cards supplied by the users. It is revealed that only 72.4%, 21.4% and 6.1% of Master, Visa and Verve credit cards respectively passed Luhn test. The higher tokenization time for each card number is attributed to the SMS delivery delay



CCISEC

CCISEC Token:303330
Please enter the token sent to you

CCISEC Token:794665
Please enter the token sent to you
2:35 PM

Figure 11: Token Confirmation Page

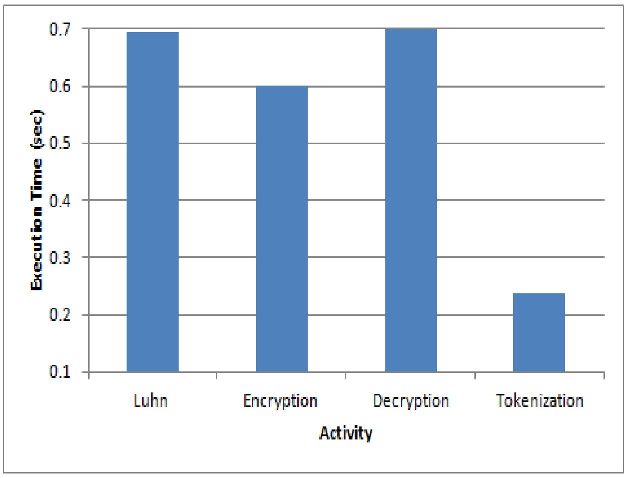


Figure 12: Activities average execution times

Table 1: Computational times

Card	Time/ frequency	Luhn Time (ms)	Time/ frequency	Encryption Time (ms)	Time/ frequency	Decryption Time (ms)	Time/ frequency	Tokenization Time (ms)
Master	104.81/223	0.47	106.797/218	0.48	106.82/218	0.49	111.18/218	0.51
Visa	30.98/66	0.46	29.90/65	0.46	30.55/65	0.47	31.85/65	0.49
Verve	7.79/19	0.41	8.17/19	0.43	7.98/19	0.42	8.74/19	0.46

The decryption time is relatively larger than the encryption time due to the variation in the length of the ciphertext generated. The encryption time for a ciphertext after performing Luhn test is relatively larger than the Luhn time and this is explained by the variation in the length of the credit card number supplied by the experimental users. The proposed system implemented 16-digit Master card, 16 or 13-digit Visa card and 19-digit Verve card. The Linkert scale-based rating of the system by users based on the metrics of reliability, speed, security, effectiveness, usability, adaptability and experience is also presented in Table 2. It is revealed that on the average, the selected users rated the system a 'very good' performance on 'reliability', 'speed', 'effectiveness', 'usability' and 'adaptability' while 'excellent' rating is recorded for 'security' and 'user experience'. These ratings gave a significant approval of the system by the selected users and also established their conviction on satisfactory performance of the system in securing credit card information during online financial transactions. Comparative analysis of the findings from the research with results from some other existing works is presented in Table 3. It is revealed that the proposed system showed better performances in securing credit card information.

Table 2: Users rating of the system

Index	Excellent (5)	V. Good (4)	Good (3)	Average (2)	Poor (1)	Mean
Reliability	850	1720	510	0	0	4
Speed	530	2020	530	0	0	4
Security	2270	790	20	0	0	5
Effectiveness	1000	1670	410	0	0	4
Usability	910	1350	810	10	0	4
Adaptability	1200	1400	460	0	0	4
User Experience	1950	890	240	0	0	5

6 CONCLUSION

This paper presented the design of an RSA encryption and tokenization-based system for combating fraud on online credit card transactions. The system requires cloud-computing technology to function and its main advantages include its ability to ensure non-repudiation of transaction as well as secrecy of card transaction data or information. Results of its implementation buttressed its effectiveness, speed efficiency and applicability. It was also revealed that the system will deliver very high usability, adaptability and favorable experience for users. Comparative analysis with related and relevant systems showed its relative advantages and superiority in areas such as credit card security, key size, mobile alert and tokenization.

REFERENCES

1 Ismail R, and Zainab A. N.: Information systems security in special and public libraries : an assessment of status, *16(2)*, 45–62 (2011).

Table 3: Comparative Analysis with other models

Metrics	**Tribhuwan *et al.*[18]**	**Kalpana & Singaraju [27]**	**Gandhi *et al.*[9]**	**Hamidi, *et.al.* [10]**	**Yang, *et al.*[19]**	**Lin, *et al.*[15]**	**Kartit, & Marraki [11]**	**Current Research**
Security Level	Average	Average	Average	Low	Average	Average	High	High
Efficiency	Average	Average	Average	Low	Low	Average	High	High
Cryposystem Algorithm	Not used	RSA	DES	Not used	Not used	BlowFish	AES and RSA	RSA
Key size (for data)	Not used	Weak (512 bits)	Weak (56 bits)	Not used	Not used	First 2 bits for security key	Average (1024 bits)	Strong (2048 bits)
Cloud environment	Not used	Used	Used	Used	Used	Used	Used	Used
Adaptability	Average	Average	Average	Low	Low	Average	High	High
Mobile Alert Service	Not used	Not used	Not used	Not used	Not used	Not used	Not used	Used for token notification
Token usage for data security	Token used for verification	Not used	Not used	Not used	Not used	Not used	Not used	Token used for data security

2 Akinyede R. O.: Development of a Payment System for e-Commerce and banking industry in Nigeria. A Thesis Submitted for Fulfillment of the Requirements for the Degree of Masters of Technology, Computer Science, The Federal University of Technology, Akure, Nigeria, (2005)

3 Internet Fraud Statistics Reports (2011), http://www.fraud.org/internet/intstat.htm., Accessed 23/04/2017

4 Li Y., Zhang W.: Securing credit card transactions with one-time payment scheme, *4*, 413–426 (2005). https://doi.org/10.1016/j.elerap.2005.06.002, Accessed 12/05/2016

5 Bhatla T. P.: Understanding Credit Card Fraud, Card Business Review (2003) http://www.tcs.com/0_whitepapers/htdocs/credit_card_fraud_white_paper_V_1.0.pdf, Accessed 02/06/2017

6 Shamir A.: Secureclick: A web payment system with disposable credit card numbers Lecture notes in computer science, Springer (2002)

7 Bjurling P.: Design and Implementation of a Secure In-app Credit Card Payment System, A Masters Thesis Submitted to Department of Computer Science, School of Computing at Linkopings University, Sweden (2013)

8 Dara J., Gundemoni L.: Credit Card Security and E-payment: Enquiry into credit Card Fraud in E-payment, A Master Thesis Continuation Courses in Computer and System Science Submitted to the Department of Business Administration and Social Sciences, Division of Information systems unit, Lulea University of Technology (2006).

9 Gandhi V., Bansal S., Kapoor R., Dhawan A.: Cloud computing security architecture-implementing des algorithm in cloud for data security", International Journal of Innovative Research in Engineering and Science (IJIRES), 9(2) (2013)

10 Hamidi N. A., Nafarieh A., Hamidi A., Robertson B.: Personalized Security Approaches in E-Banking Employing Flask Architecture over Cloud Environment. *Procedia - Procedia Computer Science*, *21*, 18–24 (2013). https://doi.org/10.1016/j.procs.2013.09.005, Accessed 18/09/2016

11 Kartit Z., Marraki M. E. L.: Applying Encryption Algorithm to Enhance Data Security in Cloud Storage, Laboratory of Research in Informatics and Telecommunication (LRIT), University of Mohammed V, Faculty of Sciences, Rabat, Morocco (2015)

12 Khan S. S., Scholar M. E.: Cloud Security Using Multilevel Encryption Algorithms, 5(1) (2016) https://doi.org/10.17148/ IJARCCE.2016.5116, Accessed 16/05/2017

13 Kim J. E.: A secure on-line credit card transaction method Based on Kerberos authentication protocol, A Master Thesis in Computer Science Submitted to the School of Computer Science, Howard R., Hughes College of Engineering, Graduate College University of Nevada, Las Vegas (2010)

14 Ko S. Y., Jeon K., Morales R: The HybrEx model for confidentiality and privacy in cloud computing. In Proceedings of the 3rd USENIX conference on Hot topics in cloud computing, HotCloud, Berkeley, CA, USA, USENIX Association (2011.)

15 Lin Q. T., Wang C. D., Pan J., Ling L., Lai J. H.: Local Data Security and Privacy Protection in Cloud Service Applications. *Ninth International Conference on Frontier of Computer Science and Technology*, 254–258 (2015). https://doi.org/10.1109/FCST.2015.39, Accessed 17/07/2017

16 Sadeghi A., Schneider T., Winandy M., Horst G.: Token-Based Cloud Computing, Springer-Verlag Berlin Heidelberg , 417–429 (2010)

17 Santiago G. P., Pereira A. C. M., Hirata J. R.: A modeling approach for credit card fraud detection in electronic payment services. *Proceedings of the*

30th Annual ACM Symposium on Applied Computing, 2328–2331 (2015). https://doi.org/10.1145/2695664.2695990, Accessed 23/11/2016

18 Tribhuwan M. R., Bhuyar V. A., Pirzade S.: A system for addressing the problems of data security in cloud data storage, International Conference on Advances in Recent Technologies in Communication and Computing, (ICARTCC), (2010)

19 Yang F., Hsu C. W., Chiu S. H.: An E-cash Payment Syste m on Cloud, Department of Computer Science and Information Engineering, Chaoyang University of Technology, 1–12 (2014).

20 Zhou M., Jiang Z.: Design and Implementation of Cloud Storage Security System, Applied Mechanics and Materials, 220, 2325-2329 (2012).

21 Din I., Mahmud F.: Secure electronic payment through RSA algorithm, manuscript submitted to Department of Computer Science and Engineering, RUET, Rajshahi (2014)

22 G. Saini, and N. Sharma, Triple Security of Data in Cloud Computing, International Journal of Computer Science and Information Technologies (IJCSIT), 5(4), 5825-5827 (2014)

23 Barr D.: Public Key Infrastructure, Technology and Programs Division, Cryptography and Boolean Operation, IJCSI International Journal of Computer Science Issues, 7(2) (2010)

24 Suresha R. G.: Enhancing Security in Cloud Storage using ECC Algorithm, International Journal of Science and Research, 2(7) (2013)

25 Kakkar A., Singh M. L., Bansal P. K.: Comparison of Various Encryption Algorithms and Techniques for Secured Data Communication in Multinode Network, *2*(1), 87–92 (2012).

26 Behera T. K., Panigrahi S.: Credit Card Fraud Detection: A Hybrid Approach Using Fuzzy Clustering andamp;amp;amp; Neural Network. *Second International Conference on Advances in Computing and Communication Engineering*, 494–499 (2015). https://doi.org/10.1109/, Accessed 18/02/2017

27 Kalpana P., Singaraju S.: Data Security in Cloud Computing using RSA Algorithm, International Journal of Research in Computer and Communication Technology, 1(4) (2012).