Vol. 9(34), Oct. 2019, PP. 4331-4340

Contents list available at IJMEC

International Journal of Mechatronics, Electrical and Computer Technology (IJMEC)

Journal Homepage: www.aeuso.org

PISSN: 2411-6173 - EISSN: 2305-0543

Homomorphic Security Automation in CloudMesh Enterprise Resource Planing Systems

K. C. Okafor[1], J. A. Okoye[2] and A. I. Chijoke[3]

[1]*Dept. of Mechatronics Engineering (Computer Systems & Software Dev.,), Federal University of Tech. Owerri, Nigeria.*

[2]*Dept. of Elect/Elect. Engineering, Chukwuemeka Odumegwu,Ojukwu University,Uli, Anambra State, Nigeria*

[3]*Dept. of Computer Science, Nnamdi Azikiwe University,Awka, Anambra State, Nigeria*

***Corresponding Author:** kennedy.okafor@futo.edu.ng*

Abstract

With smart process innovation in today's industries, enterprise resource planning (ERP) systems is currently been used for hyper-scale digital automation (HDA). The major issue is data security concerns. Previous works on ERP architectures discussed security risks, and contextualized security problems in the CloudMesh ERP environment while reporting the adaptable computational models used as counter-measures. However, multiple role-based privileges in these ERPs creates new challenges linked with public cloud ERPs (such as Cloud Software-as-a-Service ERPs). Also, security vulnerabilities impedes business innovation and service delivery in dynamic environment. Motivated by these concern, a robust ERP process architecture (CloudMesh ERP) is presented in this paper. The characteristic feature of CloudMesh ERP is the layered protective/authorization mechanism for securing classified data-asset from third party users. In the design, a computationally cohesive algorithm based on one-time-password (OTP) is encapsulated through an embedded multilayer security interface. The result is cloudmesh homomorphic encryption (CHE). OTP engine reliability, and overall system implementation are discussed. The proposed scheme improves ERP security and enhances the automation of production processes in both vertical and horizontal enterprises.

Keywords: *Enterprise ERP; Automated Cloud Security; Software Transformation; Layered Authentication.*

1. Introduction

1.1. Background

Till date, advanced software systems drives the core of global knowledge economy especially for both inventive and innovative societies. By automating traditional software models, control measures that facilitate information confidentiality, integrity, and availability are stored while been processed with a strong encryption algorithm. CloudMesh homomorphic encryption refers to an encryption format supports computation on cipher-texts and re-generates encrypted output in such a manner that when decrypted, it associates the result of the operations similar to plaintext execution. The aim is to enable processing of encrypted data inputs within an automated business process. Generally, homomorphic encryption (HE) is applied in securing unified tasks running different services without affecting sensitive data. For example, encrypted database querying in ERP

Article History:
Received Date: Mar. 14, 2019
Accepted Date: Jul. 12, 2019
Available Online: Oct. 01, 2019

systems, and Bitcoin split-key vanity mining are some useful applications of CHE services. It is useful in securing any public or private information retrieval schemes even in the first and second generation homomorphic cryptosystems.
Unfortunately, these are still intrinsically weak in terms of their security properties when compared to non-homomorphic schemes. In this work, a modified homomorphic encryption is employed in CloudMesh ERP. This serves as an automated security layer used for authorizing users as well as restricting privileges.

Looking at most large organizations in Nigeria today, SAP remains a very popular ERP system which comprises multiple modules that correlate to business processes such as manufacturing processes, material aquisation, sales, finance, plant/production, human resources 1]. Up till now, SAP R/3 implementation model used in the manufacturing industry has several security proposals that are undergoing improvement till date. Despite its challenges, ERP adaptation is increasing deployed and used by cutting-edge competitive organizations globally.

Gartner [2] showed that cloud platforms offer beneficial business analytics tools, but these needs to be secured against attack vectors. Several business collaboration could be carried out through cloud-based ERPs. As a result of this, extensive resource management and sharing appears feasible via cloud platforms supporting business processes [3] as found in ERP Systems.

According to [4], an ERP is a process unified system that is binded via a customizable knowleged mining system that structures and organizes enterprise assets and resources in a cost effective manner. Systems work flow processes are globally computed using ERPs [5]. Some forms of Cloud based ERP systems include: Decision Support Systems (DSS), Customer Relationship Management (CRM) and Supply Chain Management (SCM), [6]. These systems have rich component of decision support tools such as business intelligence, customer intelligence, supply chain intelligence, and business analytics. Cloud based ERP offers numerous advantages for instance, such systems optimizes business intelligence density, improves the quality of their composite information and makes multi-enterprise collaboration seamless.
With the recent concerns regarding security vulnerabilities, researchers and practitioners are now making efforts that leverage the cloud to achieve robust security architecture for ERPs. By integrating various business processes exclusively, a computationally cohesive security model is needed.

1.2. Research Contributions

Existing enterprise applications including manufacturing process ERPs, social networks, etc , implement first level/single layer security scheme which can be easily hacked to access an organizational critical assets. Since ERPs have high demands for task processing which cannot be effectively handled by un-virtualized servers, there is need for the development of a more secured application leveraging CHE for improved system performance, flexibility and accessibility with low cost backend. A literature discussion is discussed in Section 2.

2. Related Research Efforts

2.1. ERP Literature Contexts

In this section, a review perspective on ERP solutions has been carried out to highlight research gaps. Originally, Park and Kusiak [7] x-rayed the collective impasct of ERP investment globally but argued that a weak security implementation within the EPR context on organizational models will lead to disaster. For instance, FoxMeyer suffered disastrous business consequences due to the very diverse and risky characterization of ERP, [8]. This narrative forms the views obervesed in related research eeforst.
Adam and Peter [9] focused on duty separation in SAP ERP. Okezie *et al* [5] presented an ERP cloud implementation framework as a case study but the work did not take care of ERP security loop-holes. The authors in [4] presented a perspective definition of an ERP system. The work failed to provide robust security architecture as well as robust process architecture. Other related works on ERP focusing on the architectural deployment and Security vulnerabilities have been studied in [10], [11],[12] ,[13] and [14]. In most application context today, the message digest-5, (MD5) algorithm is a cryptographic hash function which creates 128-bit (16-byte) hash value, expressed in text and have a 32 digit hexadecimal number. This is used in the verification of data or file integrity in the ERP system, [16]. Conversely, the OTP is valid for only one login session or transaction on any thin client or server system. However, static passwords are vulnerable to replay attacks while this is not so for OTPs, hence, they are very reliable and robust for ERP system integration. Let's look at the literature gaps in context while modifying the initial security concepts in [11].

2.2. Research Gaps and Motivations

Most works deployed security policy but still lacks reinforced protection since their designs focused on unreliable single layer security scheme. Most recent systems have made efforts towards achieving very reliable security integration for various process models. However, ERP as an integrated software model still suffers from high failure rate, high implementation costs and lack of layered security. SAP R/3 implementation in enterprise business re-engineering needs high value-added output using a computationally layered security model. The aim of this study is to address the shortcomings observed in existing ERPs using a two layer CloudMesh Homomorphic algorithm (encrypt cypher algorithm) [17]. The proposed CHE relies on Message digest5 (MD5) and One Time Password (OTP) while exploring reliability constraints of CloudMesh OTP strength and its weighted payload.

3. Materials and Methods

3.1. Design Development LifeCycle

Computational Cohesive design approach was employed at the development phase of CloudMesh ERP. This was used to achieve a highly reliable Cloud ERP system in context. Fig.1a, and 1b shows the CloudMesh ERP contextual diagram. It comprises two reinforced security layers namely user login and ERP security product development modules. The work adopted SAP R/3 framework for the CloudMesh ERP business re-engineering, leveraging computationally cohesive OTP integrated with Hadoop MapReduce in Fig.1a. There are 6 subsystems implemented in CloudMesh depicted in Fig.1b. The subsystem of CloudMesh human resource (HR) includes recruitment, payroll, performance, HR intelligence, vacancies and trainings.

The subsystem of CloudMesh manufacturing includes products, raw materials, finished products, manufacturing schedule, and dispatch. The subsystem of CloudMesh Sales includes sales order, cashbook, billing, etc. The subsystem of CloudMesh marketing includes inventory, customers, distributors, marketing research, products and prices, advertising, and orders. These subsystems form the major departments of a typical product development organization. Fig. 2 is used for end user security validation. The layer-1 authentication of Fig.1b utilizes a generic connection framework in Fig. 2. As shown in Fig 1a, the Hadoop Map reduce architecture is made up of two distinct processing layers. Starting from the mapping stage to the reduce stage, the Map reduce process occurs in the task tracker. There are intermediate processes executing between map and reduce stages. These process carry out operations such as e shuffling and sorting of the mapper output data. A local file system stores the intermediate data. Consequently, the CloudMesh ERP leverages the MapReduce for parallel computation of huge datasets kept in Hadoop cluster as shown in Fig1a.

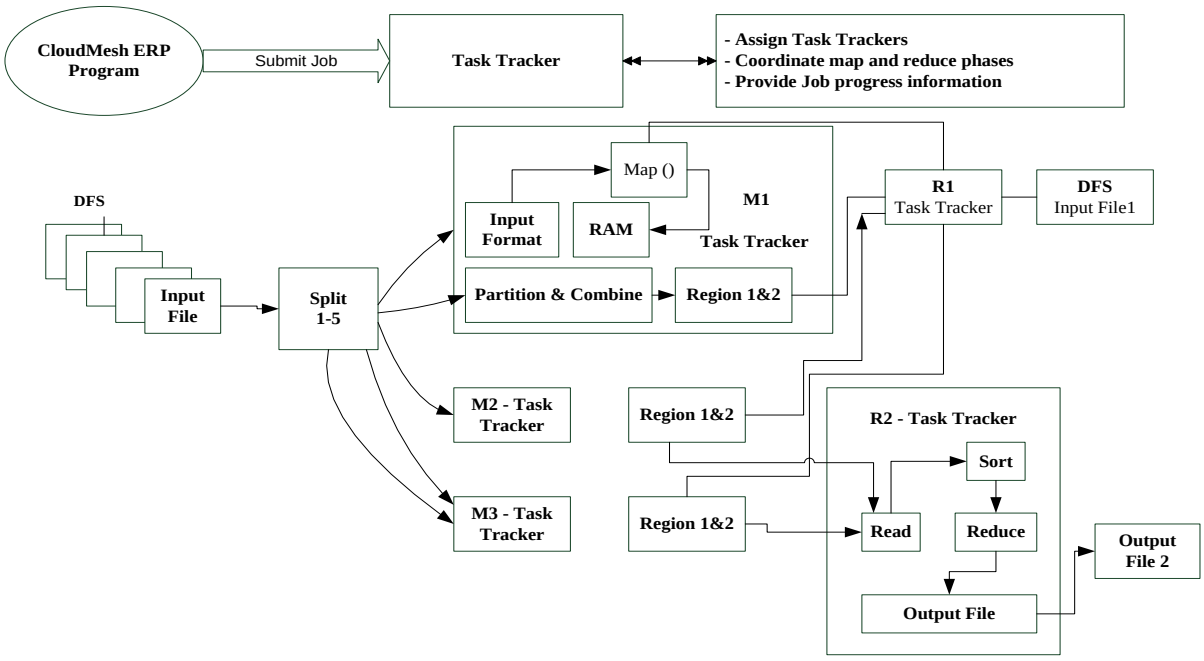

Fig. 1a: Modified CloudMesh Hadoop MapReduce Architecture.

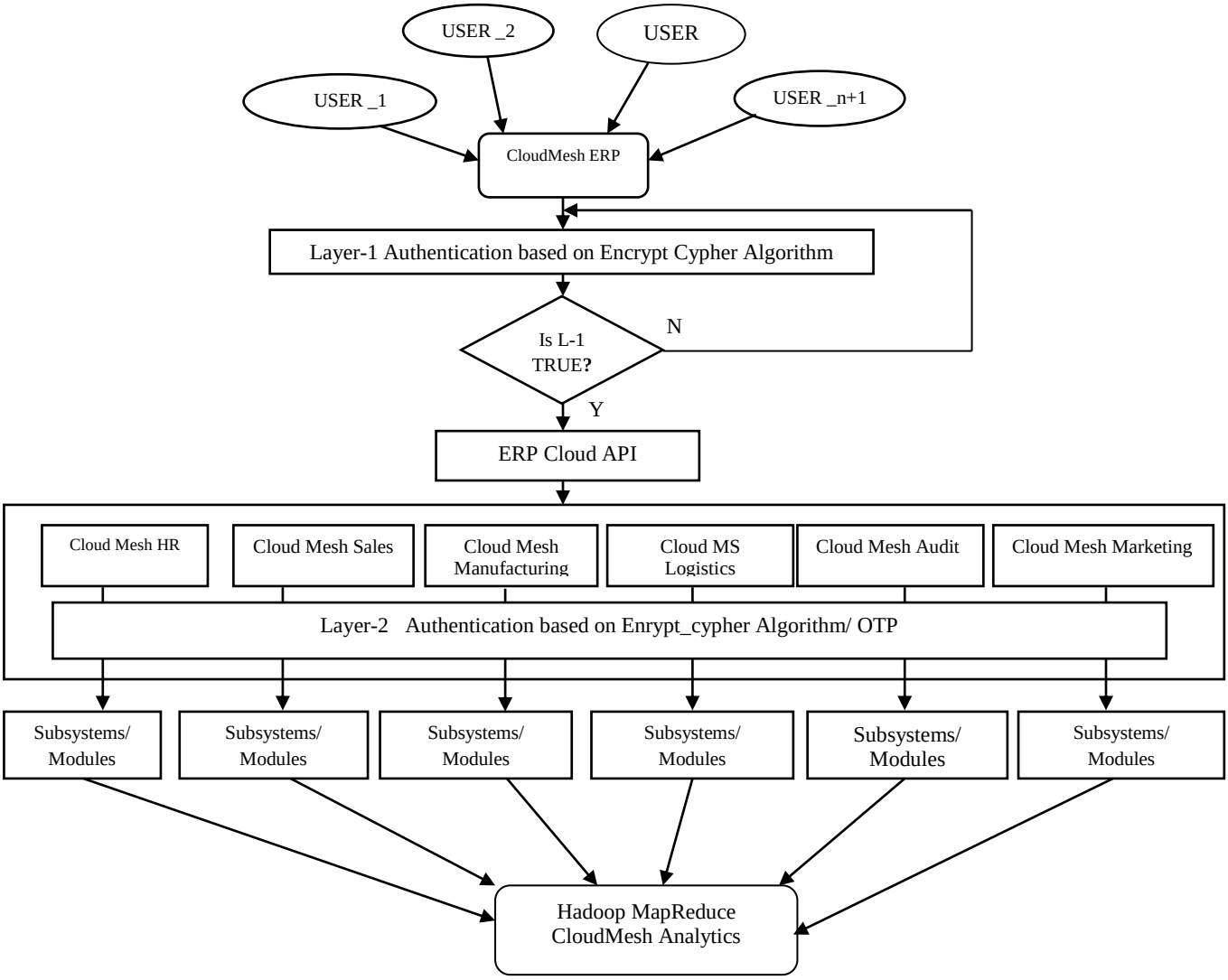

Fig. 1b: CloudMesh ERP model.

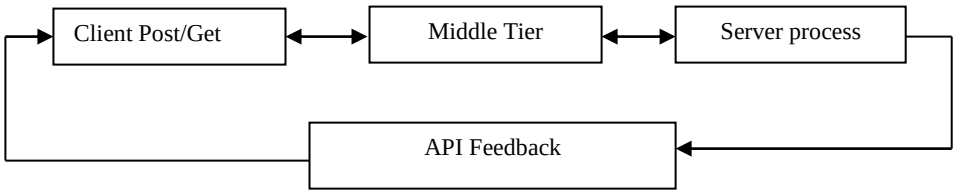

Fig. 2: User Instance Server Connection.

3.2. CloudMesh Homomorphic Encryption Algorithm

The Cloudmesh HEA relies on an already existing MD5 as expressed in an earlier work [11]. This offers an output hash value of 128-bit. Typically, this is expressed as a 32-digit hexadecimal number. The major benefit of using homomorphic encryption algorithm over other encryption algorithms is its power of customization. The user can manipulate the sequence according to demands; whether he wants it to be simple and faster or hard to crack. The algorithm in this work was designed to generate a hash value of 512-bit expressed as 128-digit hexadecimal number. The Homomorphic encryption algorithm generation consists of three major processes namely duplication process, concatenation process and interpose process. Fig.3 shows the major processes involved in encrypt cypher algorithm generation, while Fig. 4 shows the architectural framework for encrypt cypher algorithm.

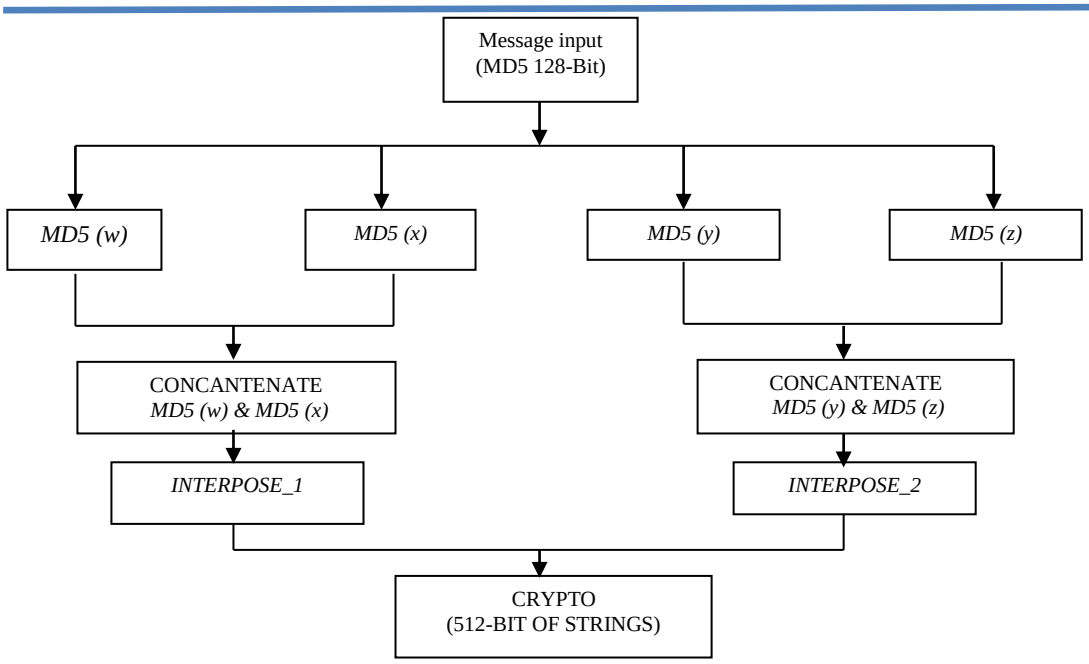

Fig. 3: Block diagram of the stages in encrypt cypher algorithm development

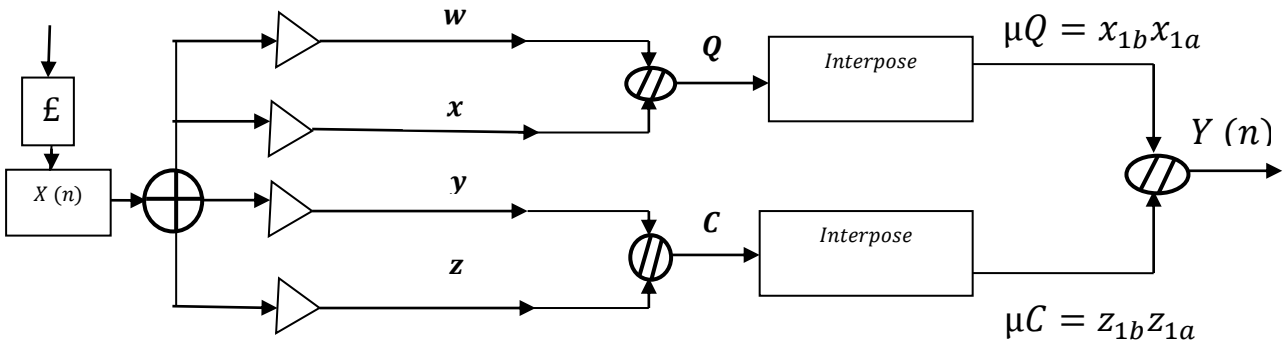

Fig. 4: A structural model for Encrypt Cypher-Algorithm.

Looking at the architectural model, the input signal (output of MD5 algorithm), which is a string of data is first acted upon by a Boolean operator known as *£*.

This operator ensures the duplication of the signal into four (*w, x, y, z*) of the same characteristics. These duplicated signals are joined together in two's as *Q* and *C* by an operator called *β*. Each of these signals are then interposed (bit swapping operation) by an operator called *μ*, which is finally joined together at the last stage to generate a 512-bit of strings or 128 hexadecimal hash value. Algorithm I shows the encrypt cypher algorithm used in the ERP system.

Algorithm I: CloudMesh Homomorphic encryption

Require:

1: *w,x,y,z*: Operator duplications
2: *β*: Joiner operator
3: *μ*: Threshold interpose operator
4: *£*: Boolean Operator
5: *f (**X**):* Objective function- Cypto 512 bits of strings

6: **procedure** INITIALIZATION

7: Class Encrypt cypher algorithm **( )**:
8: Var MD5 = system.in;
9: Varslpit = 0;
10: Var w, x, y, z;

11: **For:** slpit < 4 {
12: **Case** w:
13: MD5 (w) = MD5
14: **Case** x:
15: MD5 (x) = MD5
16: **Case y:**
17: MD5 (y) = MD5
18: **Case** z:
19: MD5 (z) = MD5}
20: **end procedure**
21: **procedure** Concatenation [*β, μ, £, f (**X**)*]
22: *μ* -Var Concat_1 => concatenate [MD5 (w), MD5 (x)];
23: *μ* -Var Concat_2 => concatenate [MD5 (y), MD5 (z)];
24: *μ* -Var Interpose_1 => interpose Concat_1;
25: *μ*- Var Interpose_2 => interpose Concat_2;
26: Var Crypto = concatenate → *£* (Interpose_1, Interpose_2);
27: *f (**X**)* -Print Crypto;
28: **end procedure**
26: **End;**

3.3. CloudMesh Homomorphic Transformation

In this Section, the mathematical characterization of CloudMesh discussed in [11] is presented. The formulation for encrypt_cypher/homomorphic algorithm is given in Equ 1.

$$Y(n) = F\{X(n)\} \quad (1)$$

Where,

$F\{.\}$ represents the encrypt cypher algorithm that acts upon the input $X(n)$ (MD5 hash value) to convert it from 128 bit of strings to 512 bit of strings. Essentially, it is a function of function that has concatenation and interposes operators embedded in it.

$X(n)$ represents input from MD5 algorithm, which is a sequence of numbers of 128-bit or 32 hexadecimal hash value.

$Y(n)$ represents the output of Encrypt_cypher algorithm, which is 512-bit of strings or 128 hexadecimal hash values.

At the first instance, the encrypt cypher algorithm acts on the input signal $\boldsymbol{X(n)}$ with n-bits to duplicate it into four signals of the same characteristics represented by w, x, y, z. This is done by introducing a duplicating operator called £, which is a Boolean OR operator. This implies that:

$$£\{X(\boldsymbol{n})\} = w + x + y + z \quad (2)$$

At the second instance, a concatenating operator β is introduced. It acts on the output components from equation (2) to concatenate the first two signals (w and x), and last two signals (y and z) as shown in equation (3) and (5).

$$\beta£\{X_1(\boldsymbol{n})\} = (w \,//\, x) \quad (3)$$

Let $Q = \beta£\{X_1(\boldsymbol{n})\}$ *and* $(w \,//\, x) = x_{1a}\, x_{1b}$, where x_{1a} and x_{1b} are the lower nibble and the upper nibble of Q respectively containing n bits. Thus,

$$Q = x_{1a}x_{1b} \quad (4)$$

$$\beta£\{X_2(\boldsymbol{n})\} = (\mathrm{y} \,//\, \mathrm{z}) \quad (5)$$

Also, let $C = \beta£\{X_2(\boldsymbol{n})\}$ and $(\mathrm{y} \,//\, \mathrm{z}) = z_{2a}z_{2b}$, where z_{2a} and z_{2b} are the lower nibble and the upper nibble of $X_2(\boldsymbol{n})$ respectively containing *n* bit, therefore

$$C = z_{2a}z_{2b} \quad (6)$$

At the third instance, an operator μ is introduced to act on both the signals from equation (4) and (6) to perform interposing (bit swapping) operation.

This implies that,

$$\mu Q = x_{1b}x_{1a} \quad (7)$$

$$\mu C = z_{2b}z_{2a} \quad (8)$$

Finally μQ and μC are concatenated, giving rise to ElGamal Cryptosystem whose homomorphic property is

$$\mathrm{Y\,(n)} = \mu Q // \mu C = x_{1b}x_{1a} // z_{2b}z_{2a} \quad (9)$$

Fig. 5 shows a block representation of the encrypt cypher algorithm high level model. The block representation is in the form of an input acted upon by a function to produce an output.

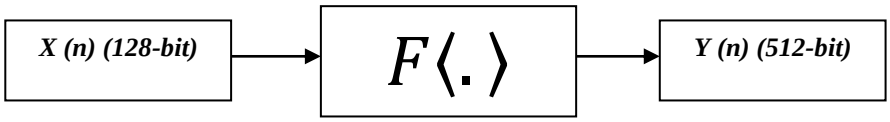

Fig 5: Encrypt cypher algorithm (high level model).

Equ. (1) is used to perform the function of modifying and increasing hash value of its input signal as shown in Fig. 5. This can be seen below.

If for example, a password is inputted into the system and the MD5 algorithm generates a code like this 110A1D1B00EE1D10, the Encrypt cypher algorithm will transform it into 00EE1D10110A1D1B00EE1D10110A1D1B00EE1D10110A1D1B00EE1D10110A1D1B.

This action is performed as follows;

Instance1: duplication of the input signal into four (4) signals of equal strength;

110A1D1B00EE1D10 110A1D1B00EE1D10 110A1D1B00EE1D10 110A1D1B00EE1D10

Instance2: concatenation takes place to give rise to;

110A1D1B00EE1D10110A1D1B00EE1D10 110A1D1B00EE1D10110A1D1B00EE1D10

Instance3: each of them is interposed as follows;

110A1D1B 00EE1D10 110A1D1B 00EE1D10

110A1D1B 00EE1D10 110A1D1B 00EE1D10

More Instance:

00EE1D10110A1D1B 00EE1D10110A1D1B

Both are combined together to give;

00EE1D10110A1D1B00EE1D10110A1D1B

Similarly, this action is performed on the second to give the same code

00EE1D10110A1D1B00EE1D10110A1D1B

Instance 4: they are both joined together to give the encrypted code;

00EE1D10110A1D1B00EE1D10110A1D1B00EE1D10110A1D1B00EE1D10110A1D1B

The coding was done using PHP. For any input provided, it generates an output of 512bits. Encrypt_cypher algorithm proffers solution to the security weakness discovered in MD5 by increasing the hash code length up to 512-bit or 128 hexadecimal value to make it stronger and harder for hackers. The presentation on OTP is discussed in Section 3.5. A reliability characterization of the OTP is presented in Section 3.4.

3.4. General Expression One-Time Password Engine Reliability

Mathematically, let the CloudMesh OTP reliability component R be given by

$$R = P(S > L) = P(S - L > 0)$$

$$= \iint f_{S,L}\,(s, L)ds\,dl \quad (10)$$

$S = Random\ nature\ of\ OTP\ strengt\square$; $L = Random\ nature\ of\ OTP\ weig\square ted\ payload$; $f_{s(\mathrm{S})}$ and $f_{l(\mathrm{L})}$ are the probability density functions of S and L

Where $f_{S,L}(S, L)$ is the joint density function of S, L. If $S, and\ L$ are independent, the interference area in OTP system gives a measure of the probability of failure state on attack.

The probability of the *L* failing in a small interval dl around l is equal to the homomorphic security area HA_l

$$p(l - \frac{dl}{2} \le L \le l + \frac{dl}{2}) = HA_l = f_{l(\mathrm{L})\mathrm{dl}} \quad (11)$$

The probability of S assuming a value larger than *L* is equal to the area under the density function of S, hence,

$$p(S > l) = HA_s \int_l^{\infty} f_{s(\mathrm{S})}\,ds = 1 - f_{S(\mathrm{l})} \quad (12)$$

If *L* and *S* are independent, the probability of the *L* value falling in the interval dl around l and the *S* value exceeding the value l occurring simultaneously gives the elemental homomorphic encryption reliability HdR_0 as

$$HdR_0 = f_{l(\mathrm{L})\mathrm{dl}} \int_l^{\infty} f_{s(\mathrm{S})}ds = f_{L(\mathrm{l})\mathrm{dl}}\,[1 - f_{S(\mathrm{l})}] \quad (13)$$

Since the homomorphic encryption reliability HdR_0 is given by the probability of *S* exceeding *L* for all possible values of *L*, this gives

$$HdR_0 = \int dR = \int_{-\infty}^{\infty} f_L\,(\mathrm{l})\left[\int_{\mathrm{l}}^{\infty} f_{s(\mathrm{S})}ds\right] dl = \int_{-\infty}^{\infty} f_{L(\mathrm{l})\mathrm{dl}}\left[1 - f_{S(\mathrm{l})}\right]dl \quad (14)$$

3.5. One-Time Password (OTP) Security Integration

In this section, the OTP integration is presented in CloudMesh ERP. Essentially, to strengthen the security in CloudMesh ERP, the OTP was integrated using soft-coding for its deployment. This involved the use of software

generated algorithm to create codes in a random manner from a predefined set of numbers using RAND function. This work deployed predefined syntaxes/functions resident in PHP, like; *Random* string and *strlen* to develop the algorithm for the OTP generation. The steps involved are as shown in Algorithm II.

Algorithm II: One Time Password Engine
1: **Start**
2: procedure Number-generation
3: Generate random numbers from 0-9
4: **Call function** → Update user table
5: **Enable** → OTP sending via a SMS Gateway
6: Verify OTP validity Status?
7: Verify Updated User table
8: End procedure
9 Ends

3.6. System Implementation

CloudMesh Implementation is presented in this Section. The system tools deployed in the CloudMesh ERP design include [11]: Hypertext Preprocessor (PHP) My Admin, XAMP Control Panel, Database Connection Bridge, SQL server, Hypertext markup Language (HTML) and CloudHub iPaaS. The developed system is a full cloud-based solution, enabling the maximization of economics and elasticity of the cloud for the integration infrastructure. Essentially, the CloudHub as a platform as a service (PaaS) component of Any point Platform, offers complete managed, multi-tenanted, globally available, secure and highly available cloud platform for integration and APIs. It offers no hardware pressure for maintenance and continuous need for software updates. Rather, it offers benefits of ERP cloud-based integration such as fully managed and hosted MuleSoft integration PaaS; multi-tenancy for applications and improved workers efficiency, globally distributed architecture that delivers 99.99% uptime, cloud security and compliance available out of the box and control access based on complex organizational requirements. The implementation of CloudMesh ERP with Encrypt cypher algorithm was achieved using PHP and HTML programming language and deployed in CloudHub PaaS. This was done by setting up the CSS8-CloudHub PaaS, enabling two major interfaces:

i. The library which comprises of MySQL server connector, ASP VBS script and XAMPP shell.
ii. Configuration files which comprises of XML and CSS connected to CloudHub.

Fig. 6 and Fig. 7 show the CloudMesh ERP layer-1 Authentication page and CloudMesh ERP HR layer-2 Authentication page respectively. The CloudMesh ERP layer-2 Authentication page which includes CloudMesh ERP; HR, Sales, manufacturing, Logistics and marketing, leads to another level of authentication using OTP as shown in Fig. 8. The developed CloudMesh ERP has been initially tested on a local host server machine running on 64bit Windows *X* operating system.

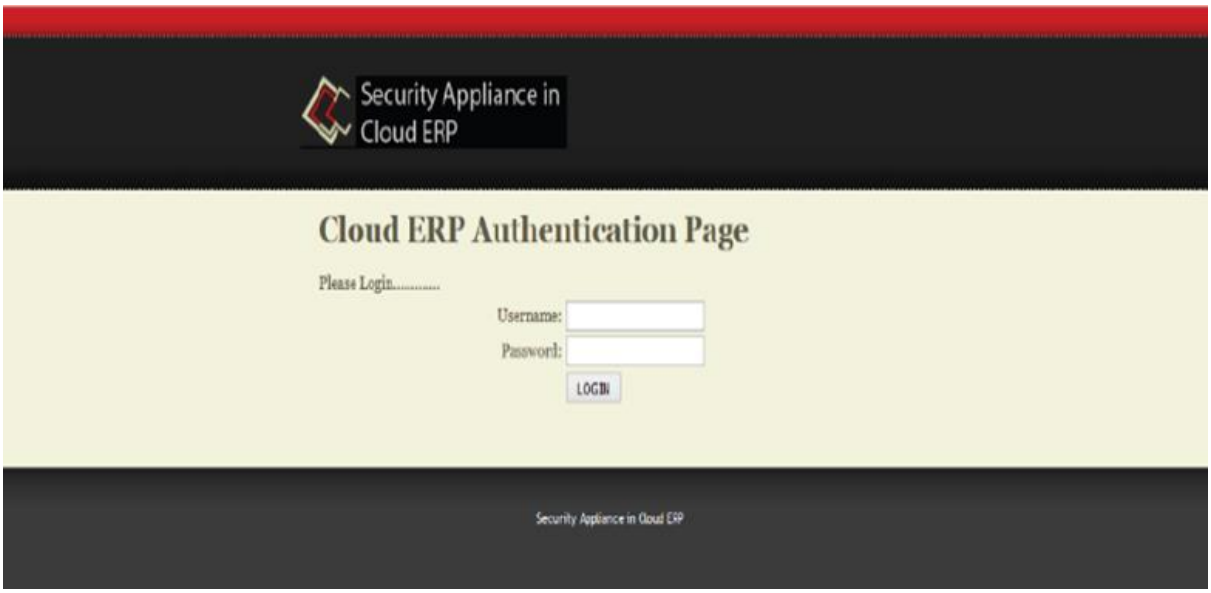

Fig. 6: CloudMesh ERP layer-1 Authentication page without login details.

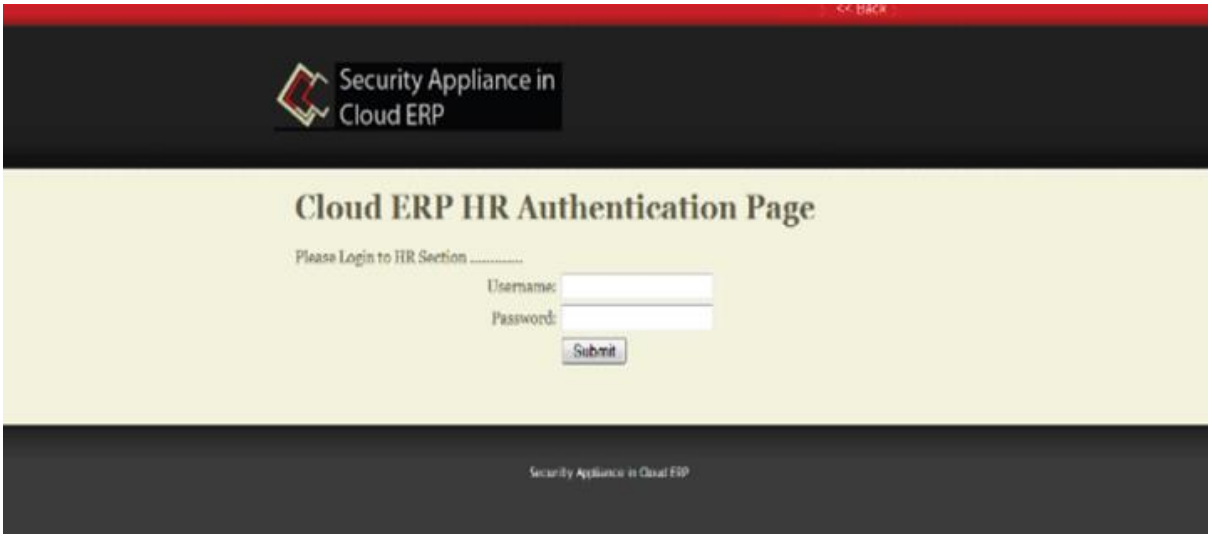

Fig. 7: CloudMesh ERP HR layer-2 Authentication page without login details.

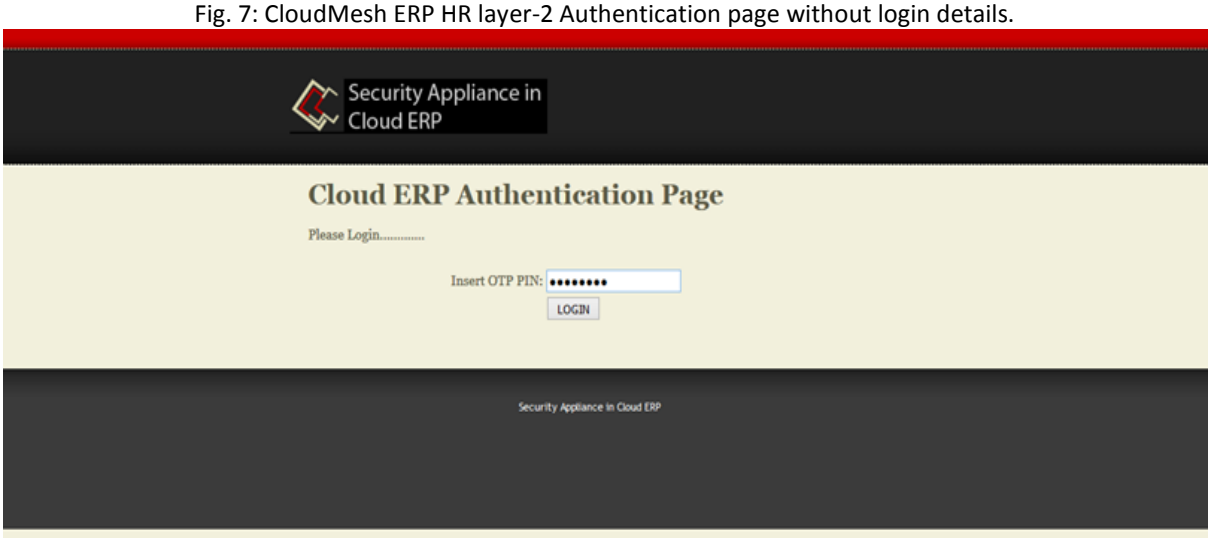

Fig. 8: CloudMesh ERP HR layer-2 Authentication page with details (OTP login).

The proposed system have its performance is similar to SAP ERP CloudHub. In accessing this application, a web browser is first launched, followed by entering the Apps URL, which on successful launch, displays the CloudMesh ERP application interface for inputting the initial username and password to access the application. On inputting the correct username and password, the internal interface housing the ERP modules is launched as shown in Fig 8. Access to any of the internal modules highlighted in Fig.1 will demand another username and password peculiar to the particular module. During input authentication, the system automatically generates a one-time password which is sent to the correct user's phone number. Another interface is displayed for the input of the received code. This serves as a way of authenticating the particular user. On the input of the correct code, the system launches or grants access to the required ERP module. However, if a wrong username and password is supplied in the first and second security layer, an error message will be displayed with room for another trial. Fig 7-8 illustrates dynamic security contexts that can be deployed immediately the compilation phase of the design is done. At compilation phase, CloudMesh ERP parses and analyzes the codes, identifies vulnerabilities, and gathers compilation reports. Any vulnerability or password mismatch after three times forms a case arbitration. An error flag is generated once vulnerability is detected during its integration/configuration. The report log is integrated into distinct management profiles. The approach used in the design is optimal and compares relatively well with the EEATP [16], and ERPs identified in previous studies.

4.CONCLUSION

A computationally intensive ERP (CloudMesh) has been developed and applied in a robust ERP process architecture. The work explored both encrypts cypher algorithm and One Time Password (OTP) to form CloudMesh Homomorphic encryption context. Hypertext Preprocessor (PHP), Hypertext Markup Language

(HTML), Structured Query Language (SQL), Apache Web server technologies and CloudHub integration were explored. The established process architecture comprises the user processes, the logic instance and the server backend logic. The secured CloudMesh ERP algorithm was deployed in order implement a multilayer (layer-2) security observed in existing models. This was designed to reduce possible vulnerabilities associated with enterprise ERPs. This type of SaaS CloudMesh ERP supports layer-2 security appliance and protects critical information assets of enterprise organizations. The system offers a more secured security algorithm with an output hash value of 512-bit expressed as a 128-digit hexadecimal number. The replacement of telvida Open Source ERP in a production environment is feasible with the proposed system.Future work will focus on using analytics to fix insider attackers and error correction at network failure for OTP reception. The elimination of crypto keys is recommended especially in complex ERPs. Also, a numerical analysis on its static and dynamic test procedures can be investigated. Finally, the application of cryptographic analytics in e-commerce ERP environment will be investigated as a novel research field.

Acknowledgement
This paper is an extended version of the paper on Enterprise Cloud Applications [11] and CloudMesh Security presented at the Faculty of Engineering International Conference (FoEIC, '18), Nnamdi Azikiwe University, Awka, Nigeria, 13th-14th, August, 2018. There is no conflict of interests.

References

[1.] SANS Institute InfoSec. Technical Aspect of Implementing/Upgrading SAP Security 4.6", 2001. Available Online: https://www.sans.org/reading-room/whitepapers/authentication/technical-aspect implementing-upgrading-sap-security-46-119.

[2.] Gartner's Top 10 Strategic Technology Trends for 2018. Available Online: https://www.gartner.com, Retrieved 1/01/2018.

[3.] Miranda, S. ERP in the Cloud: CFOs See the Value of Running Enterprise Applications As Service, *Financial Executive: Business Source Complete,* 2013, 29(1), 65-66.

[4.] She, W., & Thuraisingham, B. Security for Enterprise Resource Planning Systems. I*nformation Systems Security,* 2007, Pp.152-163.

[5.] Okezie, C., Udeze, C., & K.C. Okafor, "Cloud Computing: A Cost Effective Approach to Enterprise Web Application Implementation (A Case for Cloud ERP Web model)", *Academic Research International*, 3(1), Pp.432-433, 2012

[6.] Shafiei, F., & Sundaram, D., "Multi-Enterprise Collaborative Enterprise Resource Planning and Decision Support Systems", *Proceedings of the 37th Annual Hawaii International Conference on System Sciences,* Big Island, HI, USA., 2004, February 5-8, Retrieved from https://ieeexplore.ieee.otg/document/1265557/

[7.] Parka, K., & Kusiak, A., "Enterprise Resource Planning (ERP) Operations Support System for Maintaining Process Integration", *International Journal of Production Research*, 3959- 3982, 2005

[8.] Staehr, L,"Assessing Business Benefits from ERP Systems: An Improved ERP Benefits Framework", *Building a Better World through Information Systems*, 36, Canada: Digital Commons, 2007.

[9.] Little, A., & Best, P., "A Framework for Separation of Duties in an SAP R/3 Environment", *Managerial Auditing Journal*, 18(5), 419-430, 2003. doi.org/10.1108/02686900310476882

[10.] Ganesh, A., Shanil, K., Sunitha, C., & Midhundas, M, "Open ERP/Odoo - An Open Source Concept to ERP Solution", *In IEEE 6th International Conference on Advanced Computing (IACC)*, 2016**,** Bhimavaram, India.

[11.] Okafor, K.C, Nwafor, M., Ugwoke, F.N, & Udeze C.C, "Novel Security Integration for Vulnerability Avoidance in Enterprise Cloud Applications (CLOUD ERP)", *African Journal of Computing & ICT, 6*(5), 219-230, 2006.

[12.] Okafor, K.C, & Oluwaseyi, O. ,"Adapting Enterprise Resource Planning Systems in Organizational Models: (A Case of Research Institutes In Nigeria)", *Advances in Science and Technology*, 5(2), 95-103. 2011.

[13.] Chang, B., Hsiu, F., Tsai, Y., & Chang, Y, "An In-Cloud Enterprise Resource Planning System with High Availability and Access Control Authentication", *Int'l Conf. on Information Science, Electronics and Electrical Engineering (ISEEE),* 2014, Sapporo, Japan.doi: 10.1109/InfoSEEE.2014.6946172.

[14.] Lee, L., Leu, J., & Huang, Y. , "Implementation of Enterprise Resource Planning using the Value Engineering and System Dynamics Methods", *2nd Int'l Conf. on Information Science and Control Engineering (ICISCE)*, 2015, April 24-26 Shanghai, China. doi: 10.1109/ICISCE.2015.176.

[15.] Okafor, K.C., I.E Achumba, G.A Chukwudebe, G.C Ononiwu,"Leveraging Fog Computing for Scalable IoT Data Center using Spine-Leaf Network Topology", *Journal of Electrical and Computer Engineering*, Vol.2017, Article ID 2363240, Pp.1-11,. 2017.doi:10.1155/2017/2363240.

[16.] Okafor, K. C, G. C. Ononiwu, J. A. Okoye, M. U. Ndubuaku "Enterprise Energy Analytic Cloud Metering Portal for On Demand Service Provisioning", *Indian Journal of Science and Technology, (IJST). ISSN (Print)-0974-6846, ISSN (Online):0974-5645, Vol.10(36),* 2017, doi*: 10.17485/ijst/2017/v10i36/111913,Pp.1-13.*

[17.] Udeze, C. C, M. C. Nwafor, K. C. Okafor, C.C. Mbah, V. C. Chijindu 'Development of Enhanced Secured Cloud Mesh ERP Algorithm for Computationally Cohesive One-Time Password System, *in Proc., Faculty of Engineering International Conference (FoEIC, '18), Nnamdi Azikiwe University. Awka*, Nigeria, 13th-14th, Augu, 2018.