Kurdistan Journal of Applied Research (KJAR)
Print-ISSN: 2411-7684 | Electronic-ISSN: 2411-7706

Website: Kjar.spu.edu.iq | Email: kjar@spu.edu.iq

A New Asymmetric Fully Homomorphic Encryption Scheme for Cloud Banking Data

Zana Thalage Omar
Department of Computer
College of Science and Technology
University of Human Development,
Sulaimani, Iraq
zana.omar@uhd.edu.iq

Fadhil Salman Abed
Department of Information Technology
Kalar Technical Institute
Sulaimani Polytechnic University
Kalar,Iraq
fadhil.abed@spu.edu.iq

Shaimaa Khamees Ahmed
Computer Engineering
College of Engineering
University of Diyala
Diyala, Iraq
shaymaakhamees88@gmail.com

Article Info

Volume 5 - Issue 2 - December 2020

DOI: 10.24017/science.2020.2.12

Article history:

Received: 24 November 2020
Accepted: 24 December 2020

Keywords:

Fully Homomorphic Encryption, Cloud computing, Asymmetric Encryption, Large number, Banking security.

ABSTRACT

Most banks in our time still use the common traditional systems of high cost and relatively slow, we are now in the era of speed and technology, and these systems do not keep pace with our current age, so saving cost and time will be considered a fantastic thing for banks. The way to that is to implement cloud computing strategies with Considering data security and protection when it comes to using the cloud. The best solution to protect data security on the cloud is fully homomorphic encryption systems. The time it takes to encrypt and decrypt data is one of the main barriers it faces. Our current research provides a new algorithm for a publicly-keyed encryption system to keep bank data from tampering and theft when stored on the cloud computing platform, and our new system achieves fully Homomorphic Encryption, which allows mathematical operations to be performed on the encrypted text without the need for the original text. The security of the new system depends on the issue of analyzing huge integers, which reach 2048 bits, to their prime factors, which are considered almost impossible or unsolvable. A banking application has also been created that encrypts the data and then stores it on the cloud. The application allows the user to create accounts and deposits, transfer and withdraw funds, and everything related to banking matters.

1. INTRODUCTION

The world is witnessing rapid development and prosperity of cloud computing, as cloud computing allows the sharing of services such as (applications, storage, processing) with cloud users. The focus is on increasing the effectiveness of shared resources[1]. One of the services provided by the cloud is to save users 'data on the cloud, hence the challenges and difficulties facing the cloud providers begin, as it is their responsibility to protect the security of user data on the one hand, On the other hand, the user does not fully trust the cloud providers because they can access, modify, and delete user data Intentionally, this issue is an obstacle to cloud providers[2]. Another phenomenon is a problem when storing data on the cloud as data exchange has become a common phenomenon among cloud providers under the service agreement because this phenomenon occurs in the scenes where the data owner is not aware of this process and is considered a violation of privacy Security of user data, especially untrusted parties may participate in this process [3]. Many believe that the solution lies in the use of encryption methods when storing data in the cloud and certainly should not use low-level security encryption methods. On the contrary, it must use high-level encryption methods in terms of security[4]. Most of the existing encryption systems face two main challenges. The first is that The principal distributions face threats in most symmetric key encryption systems [5]. The second is that data must be decrypted to make adjustments to it. Therefore, cloud providers have the decryption key, and thus the data becomes unsafe[6]. In this paper, we focus on the second challenge, where we create an encryption algorithm that allows modifications to the encrypted data without the need to decrypt it. This type of encryption is called Homomorphic Encryption Systems (HE)[7]. The term "homomorphism" is derived from a Greek word-initially composed of two parts. "Homos," which means the same, and "Morphic" means the form, this type of encryption (HE) is used in computer science, where it can convert plain text into encrypted text and make adjustments to it without the need to decrypt it[7]. This type of encryption is done through three stages: the stage of generation of the encryption key, the stage of encryption, and the stage of decryption. There are several types of it, one that supports multiplication operations, one that supports addition operations, and these two types are called (Partial Homomorphic Encryption)[8], [9]. And one that supports multiplication and addition operations together and is called (Fully Homomorphic Encryption), which is the type that We present in this paper. It supports addition and multiplication operations on encrypted data without the need to decrypt the data. The proposed algorithm generates the encryption key as described in Section 8, the key generation part, and then the data is encrypted using the encryption key through a mathematical algorithm described in the encryption part in Section 8. The data is stored on the cloud in an encrypted form and when any modification or addition is made to the data, Amendment to it while it is in its encrypted state without the need to decrypt it, as the decryption key is owned by the owner of the data only and can decrypt the data through the mathematical equation described in the decryption part in Section 8 Encryption is an essential and necessary factor when storing data on the cloud where only the owner of the data can access the data, so the correct choice of the encryption algorithm is necessary for the cloud providers and users also where more efficiency and security accuracy is available [10].

2. STATEMENT OF THE PROBLEM

Cloud providers provide many services, including applications and storage many companies and users do not trust the providers of these services due to security concerns. Where the user does not upload his personal data to the cloud because the cloud providers are able to read and modify every bit loaded on the cloud and use it for personal purposes, and this thing does not comply with respecting the user’s privacy. Furthermore, some cloud providers still use traditional security techniques that are not secure with low-security level to protect user privacy. Some of the cloud providers have started to use high-level technologies to protect the privacy of users and the security of their data, but there remains a problem that the provider of the cloud itself is still able to access user data, and this is not safe for users. This problem can be solved when following FHE systems when storing data on the cloud where these systems can encrypt the data and store it in the cloud in an encrypted form and thus the cloud provider or others cannot see the data and use it, so the privacy of users and the security of their data are protected.

3. RELATED WORK

A symmetric encryption system was introduced to provide more data security and protect it from any serious attack in the year 2019 by [11], And about two years before that, specifically in (2017)data security problems were presented when stored in the cloud and a method was proposed to provide complete data security using AES encryption technology with the use of standard encryption 128 Bit by [12],in (2018) an encrypting system was introduced based on the Pailler algorithm that supports the addition process and on the RSA algorithm that supports the multiplication process on the encrypted data by [13], An encrypting scheme based on a pattern called asymmetric cipher padding (OAEP) was introduced with the symmetric cipher algorithm that stands for the RSA algorithm in (2018) by [14], and a completely symmetric encrypting system based on Euler's theory has been introduced and time complexity has been calculated and compared to other methods the size of an encryption key up to bits in (2018) by [15], while the size of the encryption key in our algorithm reaches more than 2048 bits and the encryption process is accomplished through more complicated and powerful mathematical equations, in (2018) a completely symmetric encrypting system was introduced on that relies the principle of changing a number from the plain text to another number using a secret key without converting On binary format then compare the result with DGHV and SDS systems by [16]. Not all banks use online banking services despite the tremendous benefits they enjoy due to the attacks they are subject to by cybercriminals. [17], [18] The authors present many attacks that occur on different components of online banking services, such as Spy_Eye Malware. Fraud and educational phishing are among the most common attacks on banking services, as these attacks steal user login confidentiality. [19]–[23] Researchers offer many possible solutions to phishing and attacks within browsers and across sites, but without these solutions fix the cloud-based environments. Also, risks related to banking services jobs were presented by researchers in [24].

4. BANK SERVICES IN THE CLOUD

Because of the limited use of cloud services by companies in various fields and banks, it has also created a strong incentive for cloud services providers to develop their services, especially security, as researchers in cloud affairs have been stimulated to intensify their research and efforts to find appropriate solutions for bank safety and information. The use of cloud services for banks is considered a dangerous matter to some extent because to this day storing data on the cloud is not considered a safe matter because when the data is uploaded to the cloud, control over customer data (such as account numbers, deposits, etc.) is lost, but on the other hand, there are many reasons makes banks and other institutions to use cloud services, whether public or private, including scalability, agility and saving many costs, but these benefits come with risks related to data security, you should consider these risks when using cloud services. This problem can be solved if cloud providers use strong encryption algorithms when storing customer data in the cloud. This risk is illustrated by US national law [25]. Most cloud service providers who use encryption algorithms require their customers to trust them and use their decryption keys when making any modification to their previously stored data. This does not fit with the principle of respecting the privacy of customer data security. Given the high costs of computers, the recent financial crisis, and current health conditions (COVID-19), Banks must reduce their information technology costs, but this should not be done at the expense of data security and integrity. All these reasons drive banks to use cloud services. This paper offers a simplified banking system for storing data on the cloud, this system relies on A New Asymmetric Fully Homomorphic Encryption Scheme, as this algorithm relies on data encryption, storage on the cloud and modification on request without the need to decrypt data and own a private secret key for customers, and thus the privacy of customer data security has been respected and therefore customer data (banks) on the cloud is encrypted and cannot be viewed Anyone who is not authorized is required. Our new algorithm is explained further in the remainder of this paper.

5. HOMOMORPHIC ENCRYPTION CATEGORIES

There are three main categories of Homomorphic encryption schemes: Partially Homomorphic Encryption PHE, Somewhat Homomorphic Encryption SWHE, and Fully Homomorphic Encryption FHE schemes. PHE schemes, such as RSA [8], ElGamal [26], Paillier [9], Etc., allow to applying either addition or multiplication on encrypted data. G. Kalpana et al. [27], allowing unlimited additions and a single multiplication. Construction of scheme supporting both operations addition and multiplication simultaneously is possible in 2009 by Gentry [28] by using fully homomorphic encryption.

5.1 Partially Homomorphic Encryption (PHE)

An encryption technique is called a Partially Homomorphic Encryption (PHE) if it applies only one operation on encrypted data, i.e., either addition or multiplication but not both [29].

5.2 Somewhat Homomorphic Encryption (SWHE)

The scheme that supports a limited number of homomorphic operations known as somewhat homomorphic encryption [30]. An encryption technique is called Somewhat Homomorphic encryption (SWHE) if it performs a limited number of addition and multiplication operations on encrypted data.

5.3 Fully Homomorphic Encryption (FHE)

An encryption technique is called Fully Homomorphic (FHE) if it performs both addition and multiplication simultaneously and can compute any operation [6].

6. PROPERTIES OF HOMOMORPHIC ENCRYPTION

6.1 Additive Homomorphic Encryption:

A homomorphic encryption is additive if:

$Enc\ (m1 \oplus\ m2)\ =\ Enc\ (m1)\ \oplus\ Enc\ (m2).(1)$

6.2 Multiplicative Homomorphic Encryption:

A homomorphic encryption is multiplicative, if:

$Enc\ (m1 \otimes\ m2)\ =\ Enc\ (m1)\ \otimes\ Enc\ (m2).(2)$

7. FERMAT AND EULER THEOREMS

Two important theorems presented the first by Pierre de Fermat and the second by Leonhard Euler. Both theorems are related to powers in modular arithmetic.

Fermat’s Little Theorem

Suppose that p is prime and gcd (a, p) = 1 (or a and p are relatively prime or p does not divide, then

$M^{p-1} \equiv 1 \pmod p$ (3)

7.1 Euler’s Theorem

Euler’s Theorem is a generalize of Fermat’s Little Theorem. Suppose n be an arbitrary positive integer, ø(n) denote the number of integers 1 =< a <= n such that if gcd(a, n) = 1,

then: $M^{ø(n)} \equiv 1 \pmod n$ (4)

So that:

$M^{r* ø(n) +1} \equiv M \pmod n$, when r is an integer, M<n and n=p*q where p and q are two primes number.

8. PROPOSED FULLY HOMOMORPHIC ENCRYPTION SYSTEM

8.1 The proposed scheme works as follows:

Generating the encryption key and then encrypting the numbers and texts and storing them in encrypted form on the cloud. In our work, we use a local cloud and experiment with the proposed scheme on it. The purpose of this process is to save the data encrypted on the cloud so that no one can view the data and use it for personal purposes, Therefore, when the data owner needs to perform an amendment of the encrypted data on the cloud, an encrypted request is sent to the server and the server performs mathematical operations on the encrypted data and returns an encrypted result where this encrypted result can only be decrypted through the private encryption key which is with the owner of Data only so that he can decrypt the encrypted result and see his data. In this way, we have maintained the privacy and security of the data when stored in the cloud. These procedures go through three stages. Generation the encryption key stage, the encryption stage, and the decryption stage.

8.2 Algorithm of A New Asymmetric Fully Homomorphic Encryption

- **Key Generation:**

  Generate two large Prime number p, q
  Select two big random integer z and w
  Compute $n = p * q$ and $ø\,(n) = (p-1)*(q-1)$
  Calculate $= n * z$, $ek = w * ø\,(n)$ and $Pubkey = Pk * ek$
  The public key is (Pub_{key},ek) and private key is (p, q)

- **Messages Encryption:**

  The message M will always be less than $p_{k,}$ that (m_1&m_2), (m_1+m_2) and (m_1*m_2) < P_k
  The schema of message encryption is:
  $\boldsymbol{C = Pub_{key} + m^{e_{k+1}}\ mod\ P_k}$ (5)
  Which depended only public-key (Pub_{key}, ek, Pk) to send the message
  Where
  M: Plain-text(Message), C: cipher-text

- **Message Decryption:**

  The schema of cipher decryption is:
  $M = C\ mod\ n$ (6)

8.3 Evaluate of fully Homomorphic Encryption Properties

8.3.1 To Proof correctness of the scheme:

C = Pub_{key} + M^{ek+1} mod P_k
M=C mod n = Pub_{key} + M^{ek+1} mod P_k (mod n)
= Pub_{key}(mod n) + M^{ek+1} mod n (mod P_k)
=0+ M^{ek+1} mod n (mod P_k)= M mod P_k=M (since M< P_k)

8.3.2 The Proof of Additive Homomorphic

If the following condition is fulfilled, it becomes clear to us that the proposed scheme Additive Homomorphic:

$\boldsymbol{m_1 + m_2 = dec\ [enc\ (m_1) + enc\ (m_2)]}$

Where dec is the decryption function and enc is the encryption function
Proof:

$c_1 = Pub_{key} + {m_1}^{e_{k+1}}(mod\ \ P_k) = P_k * e_k + {m_1}^{e_{k+1}}(mod\ \ P_k)$
$c_2 = Pub_{key} + {m_2}^{e_{k+1}}(mod\ \ P_k) = P_k * e_k + {M_2}^{e_{k+1}}(mod\ \ P_k)$
$c_1 + c_2$=[$P_k * e_k + {m_1}^{e_{k+1}}(mod\ \ P_k)$+ $P_k * e_k + {m_2}^{e_{k+1}}(mod\ \ P_k)$] mod n
=$P_k * e_k\ (mod\ n)$+ ${m_1}^{e_{k+1}}(mod\ \ n)$ mod P_k+$P_k * e_k\ (mod\ n)$+
${m_2}^{e_{k+1}}(mod\ \ n)$ mod P_k = m_1+m_2
[${m_1}^{e_{k+1}}(mod\ \ n)$mod P_k =m_1 mod $P_k = [m_1]$,by Fermat's Little Theorem

($m_1{}^{r*\theta(n)+1}$ mod n ≡ m_1)

$[P_k * e_k\ (mod\ n) = 0]$, since P_k =n *z, which is multiple of n modular n equal to zero.
And so on the same for other terms.

8.3.3 The Proof of Multiplicative Homomorphic

If the following condition is fulfilled, it becomes clear to us that the proposed scheme Multiplicative Homomorphic:

m_1*m_2=dec [enc (m_1) * enc (m_2)]

Where dec is the decryption function and enc is the encryption function

Proof:

m_1*m_2=dec [enc (m_1) * enc (m_2)]

$c_1 = Pub_{key} + {m_1}^{e_{k+1}}(mod\ \ P_k) = P_k * e_k + {m_1}^{e_{k+1}}(mod\ \ P_k)$

$c_2 = Pub_{key} + {m_2}^{e_{k+1}}(mod\ \ P_k) = P_k * e_k + {M_2}^{e_{k+1}}(mod\ \ P_k)$

$c_1 * c_2 = [{P_k}^2 * {e_k}^2 + P_k * e_k * {m_2}^{e_{k+1}}(\ mod\ \ \ P_k) +$

$P_k * e_k * {m_1}^{e_{k+1}}(\ mod\ \ \ P_k) + {m_1}^{e_{k+1}}(mod\ \ P_k) * {m_2}^{e_{k+1}}(mod\ \ P_k)]$ mod n

$= {P_k}^2 * {e_k}^2(mod\ n) mod\ \ P_k + P_k * e_k * {m_2}^{e_{k+1}}(\ mod\ \ \ n)$ mod P_k

$+ P_k * e_k * {m_1}^{e_{k+1}}(\ mod\ \ \ n)$ mod P_k

$+ {m_1}^{e_{k+1}}(mod\ \ \ n)$ mod $P_k * {m_2}^{e_{k+1}}(mod\ \ n)$ mod P_k =m_1*m_2

$[{m_1}^{e_{k+1}}(mod\ \ n)$mod P_k =m_1 mod $P_k = [m_1]$,by Fermat's Little Theorem($m_1{}^{r*\theta(n)+1}$ ≡ m_1)

$[P_k * e_k\ (mod\ n) = 0]$, since P_k =n *z, which is multiple of n modular n equal to zero.
And so on the same for other terms.

9. RESULTS AND DISCUSSION

Our proposed method has been applied in Java Language on a laptop that has these characteristics Intel (R) core (TM) i7-8550U CPU @1.80GHz 2.00GHz, 8 GB Ram, 64-bit Operating System, x64-based processor, Windows 10 and Big Integer library of java is used.

9.1. Case studies

In this section, several studies will be presented that we conducted to test our system to prove the creation of the secret key and its use for encryption and decryption

Case study 1:

Let us choose two different number m_1= 4, m2 = 8, select prime numbers p=467, q=307, select random number r=401, z=271 and w=449 and compute n, ø (n) =(p-1) *(q-1), pk,ek and Pub_{key} where n = p*q , pk=n*z, ek=w* µ(n) and Pub_{key} = pk*ek, as in figure 1, so n = 57490969 ,
ø (n) = 57038400, pk=15580052599, ek=25610241600 and Pub_{key} = 399008911201097918400 now we will compute c_1, c_2 where

c_1= Pub_{key} +$m_1{}^{ek+1}$ mod pk

c_1= 399008911201097918400+$4^{25610241600+1}$ mod 15580052599

c_1= 399008911209721563754

c_2= Pub_{key} +$m_2{}^{ek+1}$ mod pk

c_2= 399008911201097918400+$8^{25610241600+1}$ mod 15580052599

c_2= 399008911201097918408

A. Check the Additive Homomorphism

Let us define C_3 is the result of c_1+c_2

c_3= c_1+c_2

c_3= 399008911209721563754+ 399008911201097918408

c_3= 798017822410819482162

m_3= c_3 mod p

m_3= 798017822410819482162 mod 467

m_3= 12, which is the same of m_1+m_2 = 4 + 8 =12

B. Check the Multiplication Homomorphism

Let us define C_4 is the result of C_1*C_2

$c_4= c_1*c_2$

$c_4= 399008911209721563754* 399008911201097918408$

$c_4= 159208111221326555237218115498200062183632$

$m_4= C_4 \bmod p$

$m_4= 159208111221326555237218115498200062183632 \bmod 467$

$m_4= 32$, which is the same of $m_1*m_2 = 4 * 8 =32$

Case study 2:

We took as an example of 2048 bit A message containing several languages: English, Kurdish, Arabic and Chinese, to indicate that our scheme works in all languages. The message was:

Fully Homomorphic encryption system is the best solution when storing data in the cloud

التشفير التماثلي التام يعد افضل حل عند تخزين البيانات على السحابة

全同态加密系统是在云中存储数据时的最佳解决方案

هومومورفيك ئينكريپشن باشترين ڕێگایه بۆ هەڵگرتنی زانیاری له كڵاودا

p=17677407741877305812974453795695829556481143458816523982107269504893610803585510188955843770872491913357240854334470245431855047584556316505504922034546424105607407902410916620400236756919204260534649057763174366965692860259311987269878821499594667240449799539516699641411229698173419572164662026446722542458421793706587492634078101967282330311599671343749142994337649438535242629501024837443457010183093814050717765383414974756676321910573311062351203699632194535963913255933743321718394914382531286601992742907837501385111059060019874775880210552918496489495700486286640640775904847389333388891812125876360091783321

q=274793242160090618962920779727200735216724785471259919111969838200015924859134454935924853852331954770493497414243613151082622491841658084180729200224053219124040485802938695841984279968668247179535396815840618204133218913577931142268145416491773923591539609233733196404783888133066336919076387688113196684533173777331881353025649329821258769563456917715059460238997716592485977521637045012934473556403644381188874739662130854917666580032800918781979508747172672324090656507275368384213216100552615957975944972170017248687813245723168819511606428924641786630037666759441213644878522262418670528191902948564080139912399

r=221704313573415863159846363027780946932749477453300762390358010519665644221871408556823205605496729990175247049993604016222672733792707046542939076114622712946457774330489616626073688345733566070072769682859018095691935755933611959518359242587010941662734822633022541709516733835696135535344097858691586073225295416801346356644268401349085223233791565552540400714669486000596164658328533548435690329537958411773177609389092175837556971577934555228950186252851946253728082395142654520633100488497098153908038228839741477345543402341336195447440852503074070578011632695922887511174829871285336692705513320415788577077679

z=27524985711000283781580017271046635601832980696109986673219318963779418102363701442691287193897733121259027100064136824329301210609187447135330014261476509498296493237895245850554092082008789012353519050155215870340283325285538501161452998756484515232778505232583247168215417350878355862333125225296815221916027089844912780553832369867433821093674360804418144757102515585490143240810049118341110300098760691874841633463626819143472531107196696936095778285805195550664851606273253308770531535158506350291222037281527123330936546263878398132239648469349585337867535450178014597005244053675435612025992973391019104165051

w=2797092808624561855413075803300912860610234474784416248887272529801399484666750703759209951503644756559344326593436488713439727941370882474185729394202

2656893012503890636615100540660869650605992502165651103906619272003234752796916825734793075507607756350581033260197597066561939143858829691043997062844545316190473157943577336533399060829618243734945215508875299888937904654088180171433472303418123873469385298620381211296024121932327344367452283768916142105548731309980467698859425343574740941094513723675441376882831853221735200060302650548940693444375653163203411053243845288187881954862018473295732927544502783547

n=107695800947270025034719431871664939824276112449422620801765831440099346598114304261842177124356250874152955846256749616497421816279482807009010274311614675744981617706981159659278884034864249353366161900441850527830863014627355949776532784746004292342193106772858086635741189742878628382470222788825482078619439466918086993822425427779973735181251139233997131033805258082880864133491921278146951435149183580569943477507311533547185588288364603592019025002488258963291569442031252109254395186706720792585518142909784365241155548844983342831893209862707802361361271386053621502117821528657453815043757055435538165122108006565510450370994278689069605630824947663310654787885129930580600551258583541393276463237096719409790300468531918013125461699424164364113733499007492384170418965550955441818287555572659713469810375140114394340626758528885282437477145212493483908743852831752602650172477854216151920735037828071666571399233093893068896685065293401056907219377231430933622052700439764141956013176800084418650701249500050502819526893394414586575334429620998424324758962569305940412654440270192275049861915401702250573853044411517392855121021291426947880412860202143331118919677011230626375006094163641104605963225042768615177326418040682111088876478115340256974349331020790512970949933781062482641111417981100607563247757924181525447891711578461378207560425293369166296101163544439397409530862792837844730182746315230788913885544817429507257108778271967617395803691552285611878431448241865911206996593778231998944257712688103536162964369458335661134147062678223781442917142973016644237982757210535239639284952383592602752329293726120327361136868710322352784471970286028331332376113647464712430525790193385404246383445537711084635007910941652934052742499228368438570471199495299896505216550756874566993133659138978629083230714640376146888667525750799641

The Message after encryption (Cipher Text):

8929590021919111851902293154391529357052636795811936975669061587250996539382231001740321331646477256733520778478120002484749463886513532178306089038239543028405632852671287707959971322128447736264618470561019664744703145334553506203596158385346733535369205864387748336883902432432325314870241397654902174712584267662290418187689873676027208635786466580431611882199972233892265537394995118087925035132970272063150254470673732569902588109283169267049670523693185153912393312929173579073420316722471845203600293460239749939692865848004001981448753794055536037412565397913390978373800211854455912862135884453189535416422486433501773920700832563868399976066383721809554820188378270499951302802917378047785474957897066525091955513736270238705075302251064767107813590477919907699613784625827003099595182929464504486913045408507958278445397409776501583072082183658451896841578308880680283346545622498065328254514875222496991222059966885435694363826276685221046844963019190841823936780762571654413839930912732981414556161366337124764581505287880141762366994590146327071125276854185469427993485364825004125278257629891067521387173100110926607409684298273013314090684254325940688278657528837059993011793661294595357132479813986676562274668255571132508113770654369759346831687920038271029897667892423374810263220958410482119998979301066313625812952559462518048040949176601826723373638312781032903723117484364346218732425809651162017170527884952884907251529942128878946577168 1

4324859338796847459867096435537365305274979542714328914932946884037240943955054849884901788502012854967911585319237252603534567463085636877395614784177845969651084868080044774019427681680472229030197932192399642754776999565094106845232826861078483019102584596125270385553581382995876287798364039241380225535546428135007324080259252467858536957855977571160173001816724564278149168611602458235229345359005929681207134675994456756671341230135018897808486028637781232632613540216003627255615524973066720463955112018533057972420290031899979743542879664943985365924734927922050496995004574143976224500409164715995045548258805918415221785175620685234302659737380526775707280321443424858148236048863245063717120240164286879324078279951021526525909589449453547135404909327553086239224260691868986164678833953163322689297576670114211557653732496694098293514178254573167489897155700954496672669945551936877412606325026000009105406027427691474772489722860265140167310195531389055593255279214029079461334837210841557663927723446685822998162225135864218539770247362377169461300200659677780338825261485230955312462783622775162647230465432704120302081826922295665588001455150658147316060336688398028521563895596333938761220013450184150350880553594921185510472589502996007692047569736831254613398327847474835174479696247965785696847141506082723531940027178…..etc Due to the length of the encrypted text (Cipher Text), which reaches more than 530 pages, it has been truncated. Where the encrypted time was 151225 ms and the decrypted time was 157 ms. we have also tested it on text with 8KB in its size and several different Keys in terms of size, and we compared the results with the planners from in terms of velocity, we obtained the following results as shown in table 1 and figure 1

Table 1: Performance of small file size encryption and decryption measured in a millisecond with 64bit key length.

File size	Encryption time (ms)	Decryption time (ms)
8 kilobyte	444 ms	**54 ms**
12 kilobyte	517 ms	**56 ms**
20 kilobyte	1262 ms	**209 ms**
28 kilobyte	1705 ms	**385 ms**
40 kilobyte	2504 ms	**818 ms**

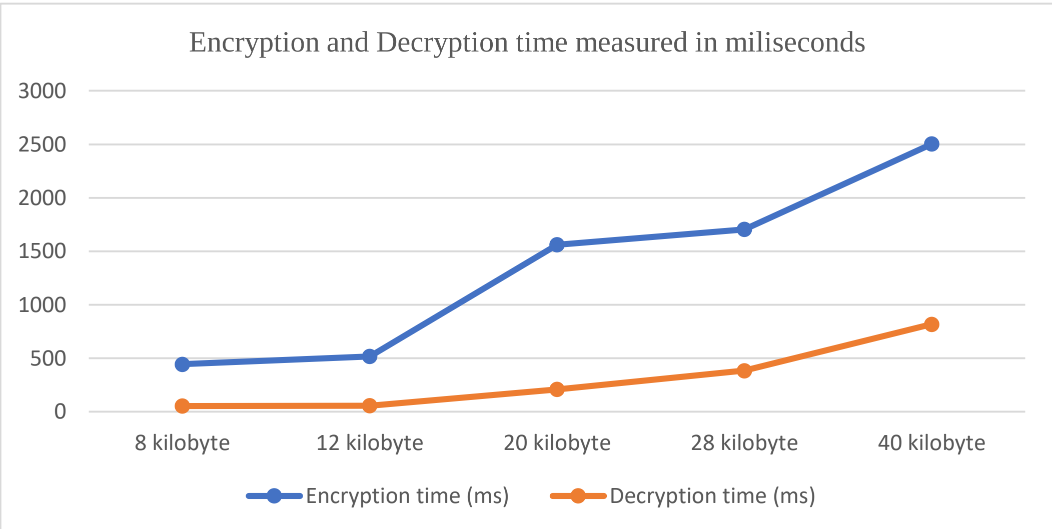

Figure 1: Computation encryption and decryption time of our schema

.9.2 Our Asymmetric Fully Homomorphic Encryption Scheme Applied to Banking Data in the cloud:

The Bank Application work as shown in figure 2

9.3 Cloud and banking app experiences:

As for our banking application, we created two accounts and encrypted them with a 2048-bit encryption key using our previously mentioned algorithm which required 1264 milliseconds as encryption time and stored it on a local private cloud as shown in figures 3 and 4. And also requests 65 milliseconds as the decryption time as is shown in figures 5 and 6

9.4 Results of NIST Statistical Tests on the Generated Secret Keys:

The randomness of this novel proposal is evaluated by the well-known NIST test suite[31]. Table 2 shows the test results of the proposed algorithm from the NIST statistical tests, demonstrating that the best statistical performance was obtained with this algorithm.

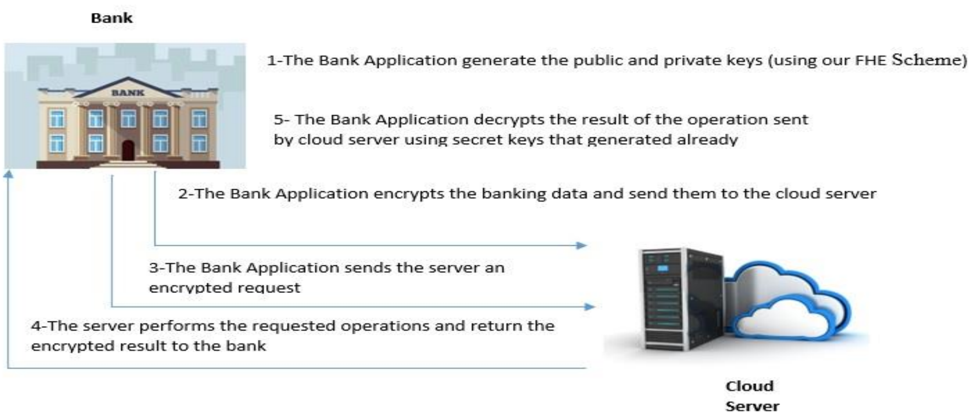

Figure 2: Scheme Representing the Link between the Bank and its Data Hosted in a Cloud Provider Server

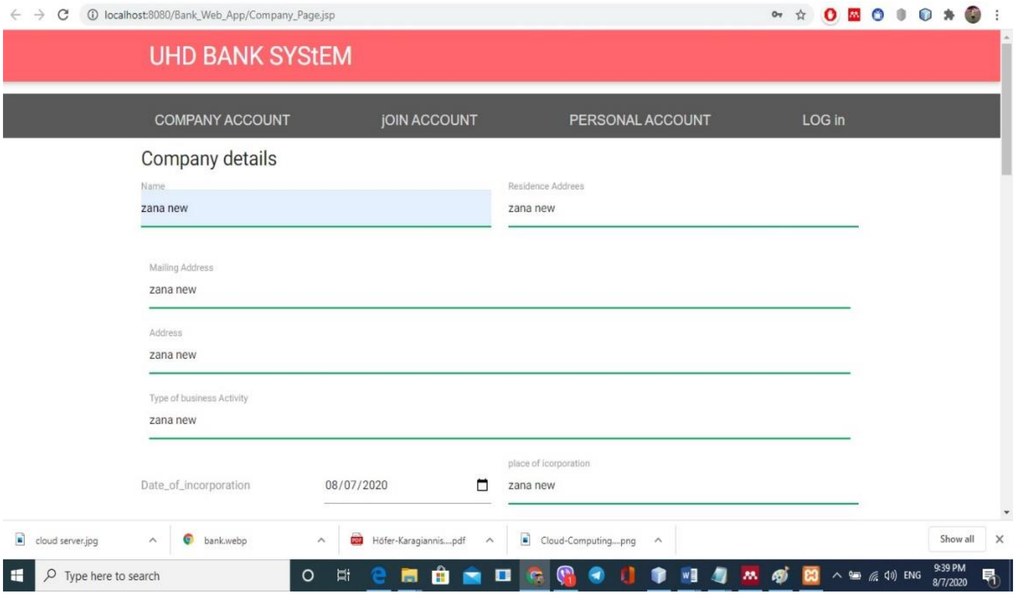

Figure 3: Create a new account and encrypt it at Bank Application Level then send to the cloud server

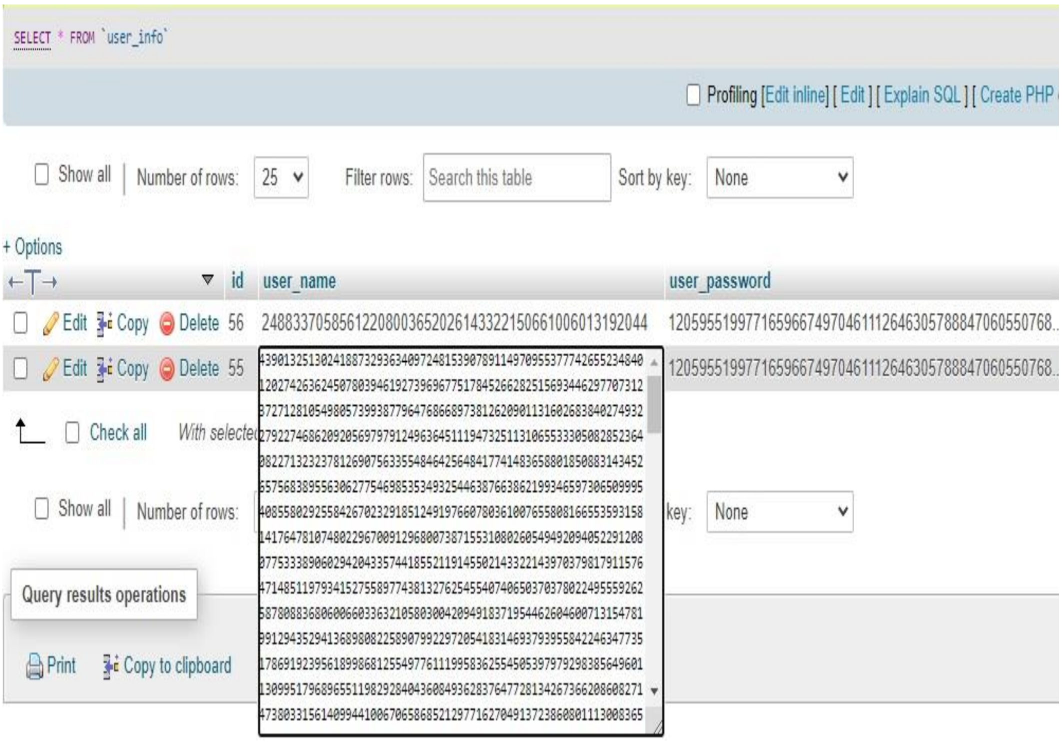

Figure 4: User login data in encrypted form at Cloud Server leve

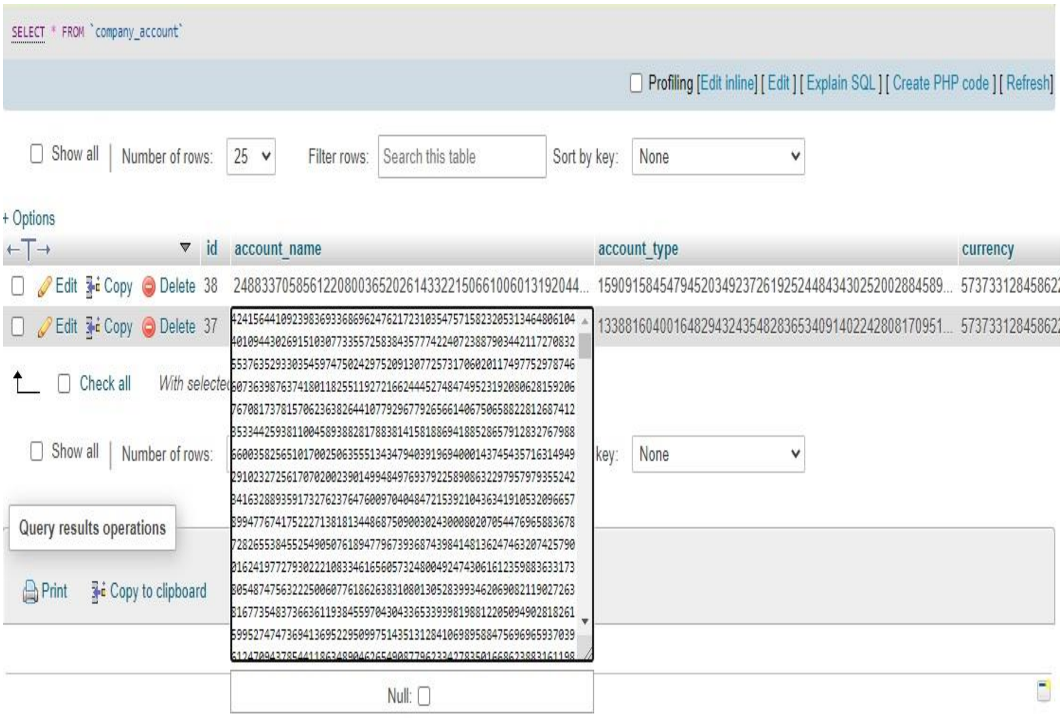

Figure 5: User Data in encrypted form at Cloud Server

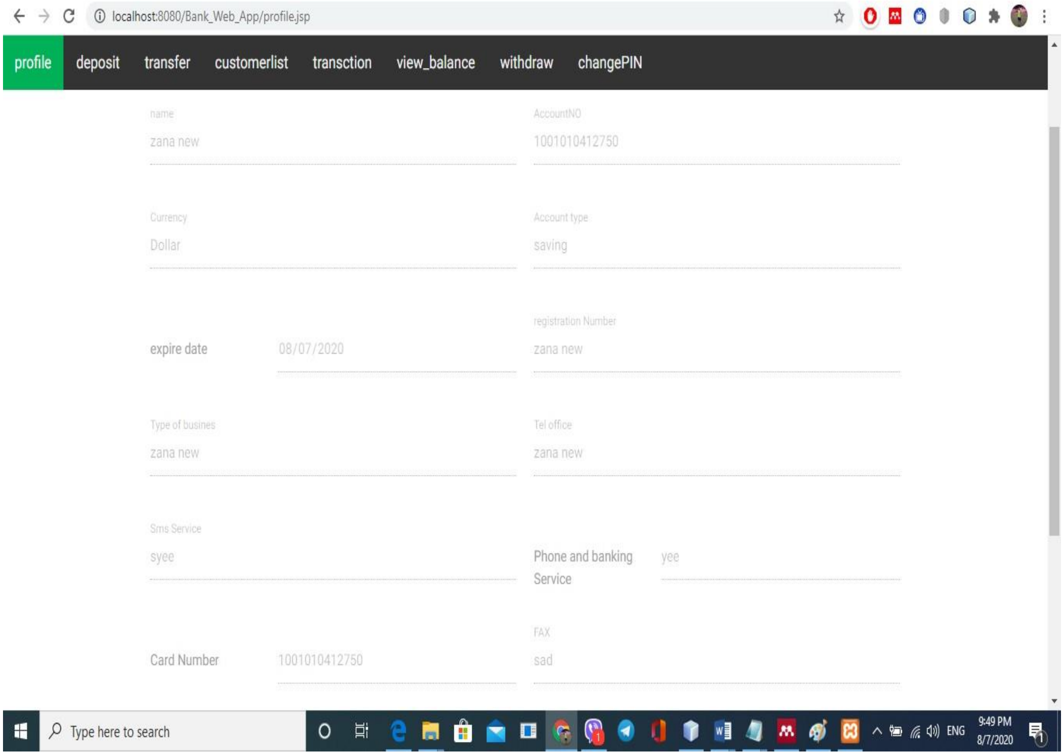

Figure 6: User Data at Bank Application Level after decrypting it

Table 2: NIST test results for the proposed algorithm

Tests	P-value	Result
Frequency (Monobits)	0.969730	SUCCESS
Block Frequency	0.934397	SUCCESS
Cumulative Sums (Cusum)	0.955178	SUCCESS
Runs	0.791649	SUCCESS
Longest Run of Ones	0.812369	SUCCESS
Rank	0.828802	SUCCESS
Discrete Fourier Transform	0.749568	SUCCESS
Non-Overlapping Template Matching	0.865923	SUCCESS
Overlapping Template Matching	0.667917	SUCCESS
Approximate Entropy	0.879027	SUCCESS
Random Excursions	0.919402	SUCCESS
Random Excursions Variant	0.942381	SUCCESS
Serial	0.842202	SUCCESS
Linear Complexity	0.909160	SUCCESS

10. CONCLUSION

In this paper, a new encryption technology based on Asymmetric(public key) Fully Homomorphic Encryption Scheme has been proposed to ensure the security of user data when stored on the cloud and respect their privacy at rest that support all languages like(English, Arabic, Kurdi and Chinese) and others, Very large prime numbers (up to 617 digits, 2048 bit) represent the strength for attack of our scheme because the proposed system depends on the problem of Factorization to the primary factors, which are considered mathematical issues under discussion at the present time and the user data is encrypted with different keys and thus provides effective security which prevents attackers from analyzing it and using it for personal purposes that ensure the security measures of the proposed technology it is resistance For any kind of brute force, mathematics, and time attacks, it explains that it can protect user data even if it is leaked to unauthorized parties, thus the proposed scheme ensures data security when it is stored in the cloud.

REFERENCE

[1] B. Varghese and R. Buyya, "Next generation cloud computing: New trends and research directions," *Futur. Gener. Comput. Syst.*, vol. 79, pp. 849–861, 2018, doi: 10.1016/j.future.2017.09.020.

[2] N. Subramanian and A. Jeyaraj, "Recent security challenges in cloud computing," *Comput. Electr. Eng.*, vol. 71, no. June, pp. 28–42, 2018, doi: 10.1016/j.compeleceng.2018.06.006.

[3] M. Masud and M. Shamim Hossain, "Secure data-exchange protocol in a cloud-based collaborative health care environment," *Multimed. Tools Appl.*, vol. 77, no. 9, pp. 11121–11135, 2018, doi: 10.1007/s11042-017-5294-5.

[4] P. R. Kumar, P. H. Raj, and P. Jelciana, "Exploring Data Security Issues and Solutions in Cloud Computing," *Procedia Comput. Sci.*, vol. 125, no. 2009, pp. 691–697, 2018, doi: 10.1016/j.procs.2017.12.089.

[5] A. Suresh and R. Varatharajan, "Competent resource provisioning and distribution techniques for cloud computing environment," *Cluster Comput.*, vol. 22, pp. 11039–11046, 2019, doi: 10.1007/s10586-017-1293-6.

[6] M. A. Mohammed and F. S. Abed, "A symmetric-based framework for securing cloud data at rest," *Turkish J. Electr. Eng. Comput. Sci.*, pp. 347–361, 2019, doi: 10.3906/elk-1902-114.

[7] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A Survey on Homomorphic Encryption Schemes," *ACM Comput. Surv.*, vol. 51, no. 4, pp. 1–35, 2018, doi: 10.1145/3214303.

[8] Rohini and T. Sharma, "Proposed hybrid RSA algorithm for cloud computing," *Proc. 2nd Int. Conf. Inven. Syst. Control. ICISC 2018*, no. Icisc, pp. 60–64, 2018, doi: 10.1109/ICISC.2018.8398902.

[9] M. E. Zhao and Y. Geng, "Homomorphic Encryption Technology for Cloud Computing," *Procedia Comput. Sci.*, vol. 154, pp. 73–83, 2018, doi: 10.1016/j.procs.2019.06.012.

[10] C. Stergiou, K. E. Psannis, B. B. Gupta, and Y. Ishibashi, "Security, privacy & efficiency of sustainable Cloud Computing for Big Data & IoT," *Sustain. Comput. Informatics Syst.*, vol. 19, pp. 174–184, 2018, doi: 10.1016/j.suscom.2018.06.003.

[11] S. Kaushik and A. Patel, "Scheme," *2019 4th Int. Conf. Internet Things Smart Innov. Usages*, pp. 1–6, 2019.

[12] M. P. Babitha and K. R. R. Babu, "Secure cloud storage using AES encryption," *Int. Conf. Autom. Control Dyn. Optim. Tech. ICACDOT 2016*, pp. 859–864, 2017, doi: 10.1109/ICACDOT.2016.7877709.

[13] X. Song and Y. Wang, "Homomorphic cloud computing scheme based on hybrid homomorphic encryption," *2017 3rd IEEE Int. Conf. Comput. Commun. ICCC 2017*, vol. 2018-Janua, pp. 2450–2453, 2018, doi: 10.1109/CompComm.2017.8322975.

[14] D. Das, "Secure cloud computing algorithm using homomorphic encryption and multi-party computation," *Int. Conf. Inf. Netw.*, vol. 2018-Janua, pp. 391–396, 2018, doi: 10.1109/ICOIN.2018.8343147.

[15] S. S. Hamad and A. M. Sagheer, "Fully Homomorphic Encryption based on Euler's Theorem," *J. Inf. Secur. Res.*, vol. 9, no. 3, p. 83, 2018, doi: 10.6025/jisr/2018/9/3/83-95.

[16] S. S. Hamad and A. M. Sagheer, "Design of fully homomorphic encryption by prime modular operation," *Telfor J.*, vol. 10, no. 2, pp. 118–122, 2018, doi: 10.5937/telfor1802118H.

[17] K. J. Hole, V. Moen, and T. Tjostheim, "Case study: Online banking security," *IEEE Secur. Priv.*, vol. 4, no. 2, pp. 14–20, 2006, doi: 10.1109/MSP.2006.36.

[18] C. Ronchi, A. Khodjanov, M. Mahkamov, and S. Zakhidov, "Security, privacy and efficiency of internet banking transactions," *World Congr. Internet Secur. WorldCIS-2011*, pp. 216–222, 2011, doi: 10.1109/worldcis17046.2011.5749854.

[19] M. Ystenes, "ET gone phishing," *New Sci.*, vol. 217, no. 2903, p. 33, 2013, doi: 10.1016/S0262-4079(13)60368-1.

[20] J. Zhan and L. Thomas, "Phishing detection using stochastic learning-based weak estimators," *IEEE SSCI 2011 Symp. Ser. Comput. Intell. - CICS 2011 2011 IEEE Symp. Comput. Intell. Cyber Secur.*, pp. 55–59, 2011, doi: 10.1109/CICYBS.2011.5949409.

[21] S. Ranjan and E. Knightly, "High performance distributed Denial-of-Service resilient web cluster

architecture," *NOMS 2008 - IEEE/IFIP Netw. Oper. Manag. Symp. Pervasive Manag. Ubiquitous Networks Serv.*, pp. 1019–1024, 2008, doi: 10.1109/NOMS.2008.4575272.

[22] F. Bin Mat Nor, K. Abd Jalil, and J. L. Ab Manan, "An enhanced remote authentication scheme to mitigate man-in-the-browser attacks," *Proc. 2012 Int. Conf. Cyber Secur. Cyber Warf. Digit. Forensic, CyberSec 2012*, pp. 271–276, 2012, doi: 10.1109/CyberSec.2012.6246086.

[23] F. Kerschbaum, "Simple cross-site attack prevention," *Proc. 3rd Int. Conf. Secur. Priv. Commun. Networks, Secur.*, pp. 464–472, 2007, doi: 10.1109/SECCOM.2007.4550368.

[24] D. LeBlanc and R. Biddle, "Risk perception of internet-related activities," *2012 10th Annu. Int. Conf. Privacy, Secur. Trust. PST 2012*, pp. 88–95, 2012, doi: 10.1109/PST.2012.6297924.

[25] A. R. Anggraini and J. Oliver, "済無No Title No Title," *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2019, doi: 10.1017/CBO9781107415324.004.

[26] E. R. Arboleda, "Secure and fast chaotic el gamal cryptosystem," *Int. J. Eng. Adv. Technol.*, vol. 8, no. 5, pp. 1693–1699, 2019.

[27] G. Kalpana, P. V. Kumar, S. Aljawarneh, and R. V. Krishnaiah, "Shifted Adaption Homomorphism Encryption for Mobile and Cloud Learning," *Comput. Electr. Eng.*, vol. 65, pp. 178–195, 2018, doi: 10.1016/j.compeleceng.2017.05.022.

[28] C. Gentry, "A Fully Homomorphic Encryption Scheme," *Dissertation*, no. September, p. 169, 2009, doi: 10.1145/1536414.1536440.

[29] A. B. Alexandru, K. Gatsis, Y. Shoukry, S. A. Seshia, P. Tabuada, and G. J. Pappas, "Cloud-based Quadratic Optimization with Partially Homomorphic Encryption," vol. 9286, no. c, pp. 1–8, 2018, doi: 10.1109/tac.2020.3005920.

[30] L. Xiong, D. Dong, Z. Xia, and X. Chen, "High-Capacity Reversible Data Hiding for Encrypted Multimedia Data with Somewhat Homomorphic Encryption," *IEEE Access*, vol. 6, pp. 60635–60644, 2018, doi: 10.1109/ACCESS.2018.2876036.

[31] P. Mell and T. Grance, "The NIST definition of cloud computing. National Institute of Standards and Technology. Special Publication 800–145. 2011." 2019.