International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 14
November 2017

Secure Data Access In Cloud With Multi-User Encrypted Sql Operations

Mustafa Abdulrazzaq Radhi Almusawi,
Master of Science (Computer Science), Department of Mathematics, University College of Science, Osmania University, Hyderabad, India

ABSTRACT:

The accomplishment of the cloud database worldview is entirely identified with solid assurances as far as administration accessibility, versatility and security, additionally of information secrecy. Any cloud supplier guarantees the security and accessibility of its stage, while the usage of versatile answers for assurance privacy of the data put away in cloud databases is an open issue left to the inhabitant. Existing arrangements address some preparatory issues through SQL operations on scrambled information. We propose the primary finish design that joins information encryption, key administration, validation and approval arrangements, and that delivers the issues identified with run of the mill danger situations for cloud database administrations. Formal models portray the proposed answers for implementing access control and for ensuring classification of information and metadata. Test assessments in view of standard benchmark sand genuine Internet situations demonstrate that the proposed design fulfills likewise adaptability and execution necessities.

Keywords:-*Relational Encrypted Database, Multi-User, cloud database, CloudNaaS*
ciphertexts

INTRODUCTION

Distributed computing is a progressive figuring strategy, by which processing assets are given powerfully through Internet and the information stockpiling and calculation are outsourced to somebody or some gathering in a 'cloud'. It extraordinarily pulls in consideration and enthusiasm from both the scholarly world and industry because of the benefit, yet it additionally has no less than three difficulties that should be taken care of before going to our genuine to the best of our insight. Above all else, information privacy ought to be ensured.

The information security is not just about the information substance. Subsequent to the most alluring part of the distributed computing is the calculation outsourcing, it is a long ways sufficiently past to simply direct an entrance control. More probable, clients need to control the benefits of information control over different clients or cloud servers. This is on account of when delicate data or calculation is outsourced to the servers may illicitly examine clients' information and access touchy data, or different clients may have the capacity to derive delicate data from the outsourced calculation. Consequently the entrance as well as the operation ought to be controlled. Also, individual data (characterized by every client's properties set) is at danger since one's personality is validated in view of his data with the end goal of access control (or benefit control in this paper).

As individuals are turning out to be more worried about their character security nowadays, the personality security likewise should be ensured before the cloud enters our life. Ideally, any power or server alone ought not know any customer's

close to home data. To wrap things up, the distributed computing framework ought to be flexible on account of security rupture in which some part of the framework is bargained by assailants.

There are three main related issues behind these two problems: execution of SQL operators over encrypted data; enforcement of access control mechanisms through selective encryption strategies; design of architectures not penalizing the performance and scalability that are typical of cloud-based services. Existing proposals offer partial and separate solutions to data confidentiality and isolation. For example, architectures supporting SQL operations on encrypted data leave access control to the cloud provider or enforce it through an intermediate trusted server. Other proposed architectures solve the problem of access control without the intervention of the cloud provider, but they do not allow execution of SQL operations on encrypted data.

We propose the first architecture, called Multi-User Relational Encrypted DataBase (MuteDB), that guarantees data confidentiality by executing SQL operations on encrypted data and by enforcing access control policies through selective encryption methods. By combining these two approaches MuteDB is the only solution ensuring confidentiality of data stored in the cloud even in the worst threat scenario where legitimate database users collude with cloud provider employees. This result is achieved through an innovative model that translates access control policies related to a plaintext database into selective encryption strategies that are applied to the corresponding encrypted database. Our solution works even in dynamic scenarios, in which users and access control policies change over time, without the need to renew and redistribute user credentials. The proposed architecture is specifically designed for cloud database scenarios where multiple users can access the cloud database through the Internet possibly from different geographical areas.

Special attention in the architectural design is devoted to guarantee the same availability and scalability of a plaintext cloud database. For this reason, MuteDB does not rely on any intermediate trusted server that could become a system bottleneck and a single point of failure. Moreover, it adopts innovative solutions for guaranteeing efficient retrieval of database metadata that are stored in an encrypted form in the cloud database. We can consider MuteDB as the first architecture that allows enterprises to leverage cloud database services while achieving the same confidentiality guarantees of a traditional in-house database and the same scalability of a cloud database service.

The performance and scalability of MuteDB are evaluated through a prototype that is subject to different query workloads based on standard (TPC-C) and recently proposed (YCSB) database benchmarks. We highlight that, as a further contribution, this paper reports the first performance evaluation studies related to encrypted cloud database services in real distributed environments where the clients are geographically distributed over the Planet Lab platform. Experimental results show that MuteDB does not affect the scalability of the original cloud service, and their performances for geographically distributed clients are comparable to those of encrypted cloud database services.

A client can unscramble the ciphertext if and just if the entrance tree in his private key is fulfilled by the qualities in the ciphertext. Nonetheless, the encryption approach is depicted in the keys, so the encrypter does not have whole control over the encryption strategy.

He needs to trust that the key generators issue keys with right structures to right clients. Besides, when a re-encryption happens, the greater part of the clients in the same framework must have their private keys re-issued in order to access the re-encoded records, and this procedure causes significant issues in execution. In the multi-user encryption, ciphertexts are made with an entrance structure, which determines the encryption arrangement, and private keys are created by traits.

A client can decode the ciphertext if and just if his traits in the private key fulfill the entrance tree indicated in the ciphertext. Thusly, the encrypter holds a definitive power about the encryption strategy. Additionally, the as of now issued private keys will never be altered unless the entire framework reboots.

LITERATURE REVIEW:

In this work, we exhibit Eucalyptus - an open-source programming structure for distributed computing that actualizes what is usually alluded to as framework as an administration (IaaS); frameworks that give clients the capacity to run and control whole virtual machine occurrences conveyed over an assortment physical assets.

We diagram the essential standards of the Eucalyptus outline, point of interest imperative operational parts of the framework, and talk about engineering exchange offs that we have made with a specific end goal to permit EUCALYPTUS to be versatile, particular and easy to use on foundation ordinarily found inside scholarly settings. At long last, we give confirm that EUCALYPTUS empowers clients acquainted with existing matrix and HPC frameworks to investigate new distributed computing usefulness while keeping up access to existing, well known application advancement programming and network middleware.

Bitar, N., Gringeri, S., & Xia, T. J. (2013), research says cloud today confront a few difficulties when facilitating line-of-business applications in the cloud. Fundamental to a large number of these difficulties is the restricted backing for control over cloud system capacities, for example, the capacity to guarantee security, execution sureties or separation, and to adaptably intervene middle boxes in application organizations. In this paper, we show the configuration and usage of a novel cloud organizing framework called CloudNaaS. Clients can influence CloudNaaS to convey applications expanded with a rich and extensible arrangement of system capacities, for example, virtual system seclusion, custom tending to, administration separation, and adaptable intervention of different middleboxes.

CloudNaaS primitives are specifically executed inside the cloud framework itself utilizing fast programmable system components, making CloudNaaS very productive. We assess an OpenFlow-based model of CloudNaaS and observe that it can be utilized to instantiate a mixed bag of system capacities in the cloud, and that its execution is hearty even despite huge quantities of provisioned administrations and connection/gadget disappointments.

Ramgovind, S., Eloff, M. M., & Smith, E. (2010, August), cloud computing has raised IT as far as possible by offering the business environment information stockpiling and limit with adaptable versatile figuring preparing energy to match flexible request and supply, whilst lessening capital use. However the open door expense of the fruitful execution of Cloud registering is to successfully deal with the security in the cloud applications. Security cognizance and concerns emerge when one starts to run applications past the assigned firewall and move closer towards the general population space.

The motivation behind the paper is to give a general security point of view of Cloud processing with the expect to highlight the security worries that ought to be appropriately tended to and figured out how to understand the maximum capacity of Cloud registering. Gartner's rundown on cloud security issues, also the discoveries from the International Data Corporation venture board study in view of cloud dangers, will be examined in this paper.

Oberheide, J., Veeraraghavan, K., Cooke, E., Flinn, J., & Jahanian, F. (2008, June) research says mobile phones keep on approaching the capacities and extensibility of standard desktop PCs. Shockingly, these gadgets are likewise starting to face a large number of the same security dangers as desktops. As of now, portable security arrangements reflect the conventional desktop display in which they run identification benefits on

the gadget. This methodology is complex and asset concentrated in both processing and force. This paper proposes another model whereby versatile antivirus usefulness is moved to an off-gadget system administration utilizing various virtualized malware location motors.

Our contention is that it is conceivable to spend data transfer capacity assets to essentially lessen on-gadget CPU, memory, and force assets. We show how our in-cloud model improves portable security and decreases on-gadget programming unpredictability, while taking into consideration new administrations, for example, stage particular behavioral examination motors.

Our benchmarks on Nokia's N800 and N95 cell phones demonstrate that our portable specialists devours a request of greatness less CPU and memory while likewise expending less power in like manner situations contrasted with existing on-gadget antivirus programming.

According to Schoo, P., Fusenig, V., Souza, V., Melo, M., Murray, P., Debar, H., ... & Zeghlache, D. (2011), Cloud computing is generally considered as an appealing administration model following the clients responsibilities for venture and operations are minimized, and expenses are in immediate connection to utilization and interest.

Be that as it may, when organizing angles for circulated mists are considered, there is little backing and the exertion is frequently disparaged. The venture SAIL is tending to cloud organizing as the blend of administration for distributed computing and basic systems administration capacities between disseminated cloud assets included to enhance the administration of both. This position paper exhibits new security challenges as considered in SAIL for guaranteeing genuine utilization of cloud systems administration assets and for averting abuse.

According to Bitar, N., Gringeri, S., & Xia, T. J. (2013), Server farm and cloud architectures keep on advancing to address the needs of expansive scale multi-occupant server farms and mists. These needs are based on seven measurements: adaptability in figuring, stockpiling, and data transfer capacity, versatility in system administrations, effectiveness in asset usage, nimbleness in administration creation, cost productivity, administration unwavering quality, and security. This article concentrates on the initial five measurements as they relate to systems administration. Vast server farms are focusing on backing for a huge number of servers, exabytes of capacity, terabits every second of activity, and countless occupants.

In a server farm, server and capacity assets are interconnected with parcel switches and switches that accommodate the data transmission and multi-occupant virtual systems administration needs.

Server farms are interconnected over the wide zone system through directing and transport advances to give a pool of assets, known as the cloud. Fast optical interfaces and thick wavelength-division multiplexing optical transport are utilized to accommodate high-limit transport intra- and between datacenter. This article surveys different exchanging, directing, and optical transport innovations, and their appropriateness in tending to the systems administration needs of vast scale multi-occupant server farms.

EXISTING SYSTEM:

Existing proposals offer partial and separate solutions to data confidentiality and isolation. For example, architectures supporting SQL operations on encrypted data leave access control to thecloud provider or enforce it through an intermediate trusted server . Other proposed architectures solve the problem of access control without the intervention of the cloud provider, but they do not allow execution of SQL operations on encrypted data.

PROPOSED SYSTEM:

Property based encryption is utilizing information transferred. This is every last hub scrambled information in store. propose the first architecture, called Multi-User relaTional Encrypted Data Base (MuteDB), that guarantees

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 14
November 2017

data confidentiality by executing SQL operations on encrypted data and by enforcing access control policies through selective encryption methods. By combining these two approaches MuteDB is the only solution ensuring confidentiality of data stored in the cloud.

Our solution works even in dynamic scenarios, in which users and access control policies change over time, without the need to renew and redistribute user credentials. The proposed architecture is specifically designed for cloud database scenarios where multiple users can access the clouddatabase through the Internet possibly from different geographical areas.

Execution is the phase of the venture when the hypothetical outline is transformed out into a working framework. Hence it can be thought to be the most basic stage in accomplishing a fruitful new framework and in giving the client, certainty that the new framework will work and be compelling. The usage stage includes watchful arranging, examination of the current framework and it's requirements on execution, planning of systems to accomplish changeover and assessment of changeover techniques.

Figure Home Page

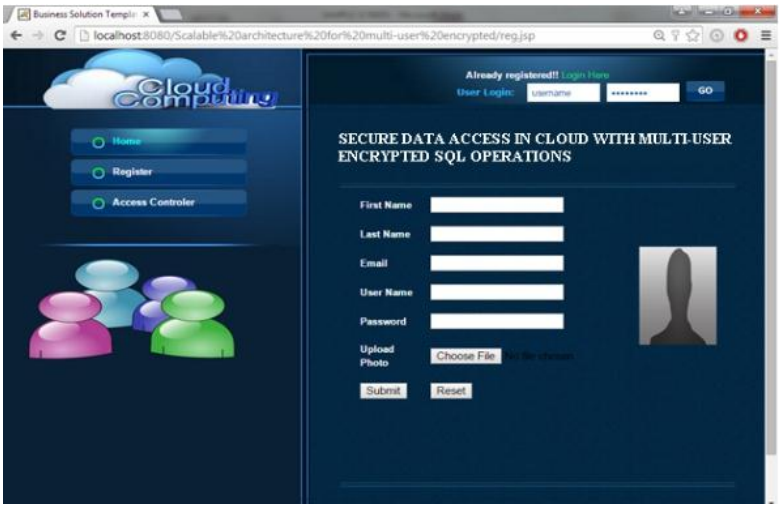

Figure Registration Page

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 14
November 2017

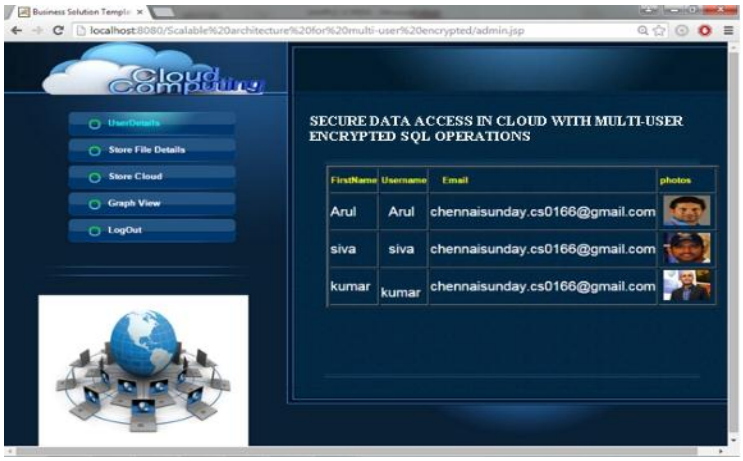

Figure User Login

Figure : Download File

RSA Algorithm

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 14
November 2017

RSA was first described in 1977 by Ron Rivest, Adi Shamir and Leonard Adleman of the Massachusetts Institute of Technology. Public-key cryptography, also known as asymmetric cryptography, uses two different but mathematically linked keys, one public and one private. The public key can be shared with everyone, whereas the private key must be kept secret. In RSA cryptography, both the public and the private keys can encrypt a message; the opposite key from the one used to encrypt a message is used to decrypt it. This attribute is one reason why RSA has become the most widely used asymmetric algorithm: It provides a method of assuring the confidentiality, integrity, authenticity and non-reputability of electronic communications and data storage.

Many protocols like SSH, OpenPGP, S/MIME, and SSL/TLS rely on RSA for encryption and digital signature functions. It is also used in software programs -- browsers are an obvious example, which need to establish a secure connection over an insecure network like the Internet or validate a digital signature. RSA signature verification is one of the most commonly performed operations in IT.

Explaining RSA's popularity

RSA derives its security from the difficulty of factoring large integers that are the product of two large prime numbers. Multiplying these two numbers is easy, but determining the original prime numbers from the total -- factoring -- is considered infeasible due to the time it would take even using today's super computers.

The public and the private key-generation algorithm is the most complex part of RSA cryptography. Two large prime numbers, *p* and *q*, are generated using the Rabin-Miller primality test algorithm. A modulus *n* is calculated by multiplying *p* and *q*. This number is used by both the public and private keys and provides the link between them. Its length, usually expressed in bits, is called the key length. The public key consists of the modulus *n*, and a public exponent, *e*, which is normally set at 65537, as it's a prime number that is not too large. The *e* figure doesn't have to be a secretly selected prime number as the public key is shared with everyone. The private key consists of the modulus *n* and the private exponent *d,* which is calculated using the Extended Euclidean algorithm to find the multiplicative inverse with respect to the totient of *n*.

A simple, worked example

Alice generates her RSA keys by selecting two primes: p=11 and q=13. The modulus $n=p\times q=143$. The totient of n $\phi(n)=(p-1)x(q-1)=120$. She chooses 7 for her RSA public key *e* and calculates her RSA private key using the Extended Euclidean Algorithm which gives her 103.

Bob wants to send Alice an encrypted message *M* so he obtains her RSA public key (*n*,e) which in this example is (143, 7). His plaintext message is just the number 9 and is encrypted into cipher text *C* as follows:

$$M^e \bmod n = 9^7 \bmod 143 = 48 = C$$

When Alice receives Bob's message she decrypts it by using her RSA private key (*d*, *n*) as follows:

$$C^d \bmod n = 48^{103} \bmod 143 = 9 = M$$

To use RSA keys to digitally sign a message, Alice would create a hash or message digest of her message to Bob, encrypt the hash value with her RSA private key and add it to the message. Bob can then verify that the message has been sent by Alice and has not been altered by decrypting the hash value with her public key. If this value matches the hash of the original message, then only Alice could have sent it (authentication and non-repudiation) and the message is exactly as she wrote it (integrity). Alice could, of course, encrypt her message with Bob's RSA public key (confidentiality) before sending it to Bob. A digital certificate contains information that identifies the certificate's owner and also contains the owner's public key. Certificates are signed by the certificate authority that issues them, and can simplify the process of obtaining public keys and verifying the owner.

International Journal of Research
Available at https://edupediapublications.org/journals
e-ISSN: 2348-6848
p-ISSN: 2348-795X
Volume 04 Issue 14
November 2017

Security of RSA

As discussed, the security of RSA relies on the computational difficulty of factoring large integers. As computing power increases and more efficient factoring algorithms are discovered, the ability to factor larger and larger numbers also increases. Encryption strength is directly tied to key size, and doubling key length delivers an exponential increase in strength, although it does impair performance. RSA keys are typically 1024- or 2048-bits long, but experts believe that 1024-bit keys could be broken in the near future, which is why government and industry are moving to a minimum key length of 2048-bits. Barring an unforeseen breakthrough in quantum computing, it should be many years before longer keys are required, but elliptic curve cryptography is gaining favor with many security experts as an alternative to RSA for implementing public-key cryptography. It can create faster, smaller and more efficient cryptographic keys. Much of today's hardware and software is ECC-ready and its popularity is likely to grow as it can deliver equivalent security with lower computing power and battery resource usage, making it more suitable for mobile apps than RSA. Finally, a team of researchers which included Adi Shamir, a co-inventor of RSA, has successfully determined a 4096-bit RSA key using acoustic cryptanalysis, however any encryption algorithm is vulnerable to this type of attack.

Diffie Hellman Algorithm

Diffie-Hellman key exchange, also called exponential key exchange, is a method of digital encryption that uses numbers raised to specific powers to produce decryption keys on the basis of components that are never directly transmitted, making the task of a would-be code breaker mathematically overwhelming.

To implement Diffie-Hellman, the two end users Alice and Bob, while communicating over a channel they know to be private, mutually agree on positive whole numbers p and q, such that p is a prime number and q is a generator of p. The generator q is a number that, when raised to positive whole-number powers less than p, never produces the same result for any two such whole numbers. The value of p may be large but the value of q is usually small.

Once Alice and Bob have agreed on p and q in private, they choose positive whole-number personal keys a and b, both less than the prime-number modulus p. Neither user divulges their personal key to anyone; ideally they memorize these numbers and do not write them down or store them anywhere. Next, Alice and Bob compute public keys a^* and b^* based on their personal keys according to the formulas

$a^* = q^a \bmod p$ AND $b^* = q^b \bmod p$

The two users can share their public keys a^* and b^* over a communications medium assumed to be insecure, such as the Internet or a corporate wide area network (WAN). From these public keys, a number x can be generated by either user on the basis of their own personal keys. Alice computes x using the formula

$x = (b^*)^a \bmod p$

Bob computes x using the formula

$x = (a^*)^b \bmod p$

The value of x turns out to be the same according to either of the above two formulas. However, the personal keys a and b, which are critical in the calculation of x, have not been transmitted over a public medium. Because it is a large and apparently random number, a potential hacker has almost no chance of correctly guessing x, even with the help of a powerful computer to conduct millions of trials. The two users can therefore, in theory, communicate privately over a public medium with an encryption method of their choice using the decryption key x.

The most serious limitation of Diffie-Hellman in its basic or "pure" form is the lack of authentication. Communications using Diffie-Hellman all by itself are vulnerable to man in the middle attacks. Ideally, Diffie-Hellman should be used in conjunction with a recognized authentication method such as digital signatures to verify the identities of the users over the public communications medium. Diffie-Hellman is well suited for use in data communication but is less often used for data stored or archived over long periods of time.

Encryption algorithm

The following steps should be followed to develop an encrypted text:

1.) Generate the ASCII value of the letter
2.) Generate the corresponding binary value of it (Binary value should be 8 digits e.g. for decimal 32 binary number should be 00100000)
3.) Reverse the 8 digit's binary number
4.) Take a 4 digits divisor (>=1000) as the Key
5.) Divide the reversed number with the divisor
6.) Store the remainder in first 3 digits & quotient in next 5 digits (remainder and quotient wouldn't be more than 3 digits and 5 digits long respectively. If any of these are less than 3 and 5 digits respectively we need to add required number of 0s (zeros) in the left hand side. So, this would be the cipertext i.e. encrypted text.

Now store the remainder in first 3 digits & quotient in next 5 digits.

Let us see an example to apply the above mentioned steps:

- Encryption

Let, the character is "T". Now according to the steps we will get the following:

- Step 1: ASCII of "T" is 84 in decimal.
- Step 2: The Binary value of 84 is 1010100. Since it is not an 8 bit binary number we need to make it 8 bit number as per the encryption algorithm. So it would be 01010100
- Step 3: Reverse of this binary number would be 00101010
- Step 4: Let 1000 as divisor i.e. Key
- Step 5: Divide 00101010 (dividend) by 1000(divisor)
- Step 6: The remainder would be 10 and the quotient would be 101. So as per the algorithm the ciphertext would be 01000101 which is ASCII 69 in decimal i.e. "E"

Decryption algorithm

- Step 1: Multiply last 5 digits of the ciphertext by the Key
- Step 2: Add first 3 digits of the ciphertext with the result produced in the previous step
- Step 3: If the result produced in the previous step i.e. step 2 is not an 8-bit number we need to make it an 8- bit number
- Step 4: Reverse the number to get the original text i.e. the plain text
- Decryption

- Step 1: Multiply last 5 digits of the ciphertext by the Key
- Step 2: Add first 3 digits of the ciphertext with the result produced in the previous step
- Step 3: If the result produced in the previous step i.e. step 2 is not an 8-bit number we need to make it an 8- bit number
- Step 4: Reverse the number to get the original text i.e. the plain text

CONCLUSION

In this paper we propose MuteDB, a novel architecture for cloud database services that guarantees for the first time data confidentiality through SQL-aware encryption algorithms and data isolation through access control enforcement based on encryption and key derivation techniques. These solutions allow MuteDB to address threat issues that are relevant for cloud services including risks of information leakage due to collusions between cloud provider employees and tenant users. The most important solutions are described through formal models, while the feasibility, performance and scalability of the proposed architecture are demonstrated through a large set of experiments carried out through a prototype deployed in a real Internet-based environment where cloud database services are accessed concurrently by geographically distributed clients.

All results confirm that for realistic workloads, the MuteDB architecture achieves performance and scalability comparable to those of unencrypted cloud database services. On going work is focused on integrating private information retrieval solutions in MuteDB with the goal of preventing information leakage caused by access pattern analyses, and novel architectural solutions for hybrid cloud environments.

REFERENCE:

[1] Oberheide, J., Veeraraghavan, K., Cooke, E., Flinn, J., & Jahanian, F. (2008, June). Virtualized in-cloud security services for mobile devices. In Proceedings of the First Workshop on Virtualization in Mobile Computing (pp. 31-35). ACM.

[2] Schoo, P., Fusenig, V., Souza, V., Melo, M., Murray, P., Debar, H., ... & Zeghlache, D. (2011). Challenges for cloud networking security. In Mobile Networks and Management (pp. 298-313). Springer Berlin Heidelberg.
[3] Bitar, N., Gringeri, S., & Xia, T. J. (2013). Technologies and protocols for data center and cloud networking. Communications Magazine, IEEE, 51(9), 24-31.
[4] Houidi, I., Mechtri, M., Louati, W., & Zeghlache, D. (2011, July). Cloud service delivery across multiple cloud platforms. In Services Computing (SCC), 2011 IEEE International Conference on (pp. 741-742). IEEE.
[5] Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., & Zagorodnov, D. (2009, May). The eucalyptus open-source cloud-computing system. In Cluster Computing and the Grid, 2009. CCGRID'09. 9th IEEE/ACM International Symposium on (pp. 124-131). IEEE.
[6] Wodczak, M. (2011, November). Resilience aspects of autonomic cooperative communications in context of cloud networking. In Network Cloud Computing and Applications (NCCA), 2011 First International Symposium on (pp. 107-113). IEEE.
[7] Bitar, N., Gringeri, S., & Xia, T. J. (2013). Technologies and protocols for data center and cloud networking. Communications Magazine, IEEE, 51(9), 24-31.
[8] Bechler, M., Hof, H. J., Kraft, D., Pahlke, F., & Wolf, L. (2004, March). A cluster-based security architecture for ad hoc networks. In INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies (Vol. 4, pp. 2393-2403). IEEE.
[9] Covington, M. J., Fogla, P., Zhan, Z., & Ahamad, M. (2002). A context-aware security architecture for emerging applications. In Computer Security Applications Conference, 2002. Proceedings. 18th Annual (pp. 249-258). IEEE.
[10] Zhou, L., & Chao, H. C. (2011). Multimedia traffic security architecture for the internet of things. Network, IEEE, 25(3), 35-40.
[11] Pensak, D. A., Cristy, J. J., & Singles, S. J. (2001). U.S. Patent No. 6,289,450. Washington, DC: U.S. Patent and Trademark Office.
[12] Nagaratnam, N., Janson, P., Dayka, J., Nadalin, A., Siebenlist, F., Welch, V., ... & Tuecke, S. (2002). The security architecture for open grid services. Open Grid Service Architecture Security Working Group (OGSA-SEC-WG), 1-31.