Data Confidentiality in Public Cloud: A Method for Inclusion of ID-PKC Schemes in OpenStack Cloud

Bhanu Prakash Gopularam
N.M.I.T
Bangalore, Karnataka, India

Nalini N
N.M.I.T
Bangalore, Karnataka, India

ABSTRACT
The term data security refers to the degree of resistance of protection given to information from unintended or unauthorized access. The core principles of information security remain the same - Confidentiality, Integrity and Availability also referred as CIA triad. With cloud adoption the confidential enterprise data is moved from organization premises to untrusted public network and due to this the attack surface has increased manifold. Several cloud computing platforms like OpenStack, Eucalyptus, Amazon EC2 offer users to build and configure public, hybrid and private clouds. While the traditional encryption based on PKI infrastructure still works in cloud scenario the management of public-private keys and trust certificates is difficult. The Identity based Public Key Cryptography (also referred as ID-PKC) overcomes this problem by using publicly identifiable information for generating the keys and works well with decentralized systems. The users can exchange information securely without having to manage any trust information. Another advantage is that access control (role based access control policy) information can be embedded into data unlike in PKI where it is handled by separate component or system. In OpenStack cloud platform the keystone service acts as identity service for authentication and authorization and has support for public key infrastructure for auth services. The proposed approach explains cloud security model using OpenStack cloud platform and analyzes its security architecture for data confidentiality. It provides a method to integrate ID-PKC schemes for securing data when in transit and storage and explains the key measures for safe guarding data. The proposed approach uses JPBC crypto library for key-pair generation based on IEEE standard(s) P1636.3 for assuring data confidentiality in public cloud environment.

General Terms
Cloud Computing, Data Confidentiality, Identity Based Cryptography, Secure Communication, Token Management

Keywords
Data Encryption, Key Policy Attribute based encryption, OpenStack keystone, Token scoping

1. INTRODUCTION
Cloud computing provides computation, software applications, data management and storage resources without requiring cloud users to know the location and other details of the computing infrastructure. Cloud computing infra-structures enable companies to cut costs by outsourcing computations on-demand. Cloud services are offered in different service models such as Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) as shown in Fig. 1. The main cloud information security objectives are dependability, trustworthiness, survivability (resilience). [U.S. DoD Software Assurance Initiatives]. Also the data confidentiality, integrity and availability also known as CIA triad are 3 important concepts of cloud software assurance [6] for information system security.

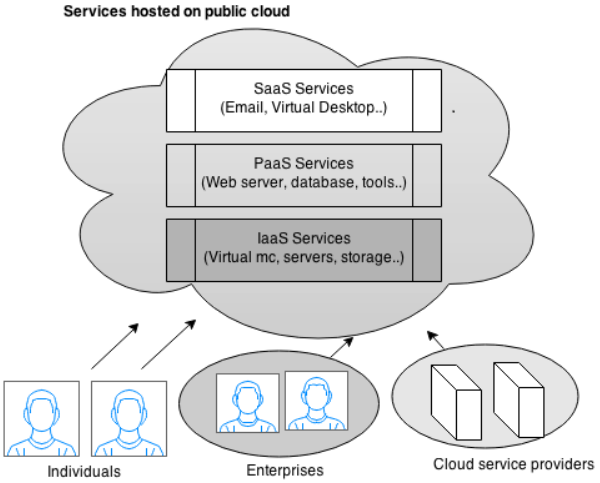

Fig 1: Cloud computing service models

Public-key cryptography refers to cryptographic algorithm where separate keys are used for encrypting and decrypting the content. The public key is used for encryption where as private key used for decryption as shown in Fig. 2. Public-key algorithms have become key ingredients for web security and it mainly serves two purposes. The receiver's public key is used for encrypting the messages and in digital signatures where the message is signed with sender's private key. The public-key infrastructure relates to hardware, software, people and policies needed to create, manage, distribute the digital certificates.

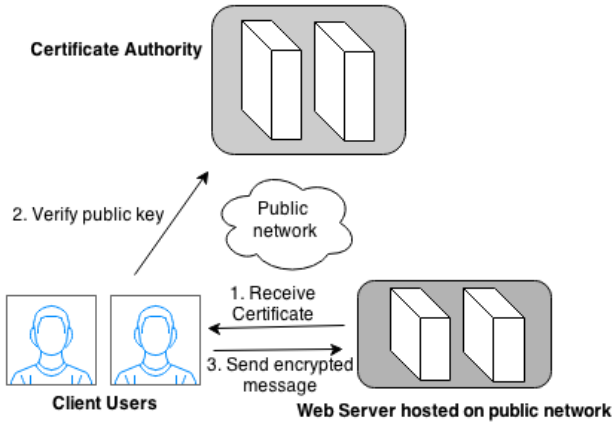

Fig 2: PKI system supporting certificate service

The certificate management is a complex task [3] in PKI where the user keys used for communication are maintained. When a particular key is revoked the corresponding message needs to be propagated to all stakeholders having the content as quickly as possible.

Identity-based cryptography is a variant of public-key cryptography in which a publicly known information such as email-id, SSID, IP address, expiry-date etc. is used in public key generation. First implementation of ID-PKC schemes was envisaged by Adi Shamir in 1984, which allowed users to verify the digital signatures using publicly available user information. Unlike PKI schemes ID-PKC system does not require certificate authority and eliminates the need for separate public key distribution infrastructure [7]. ID-PKC systems require private key generator (PKG) which holds the master private key and takes responsibility of generating the public keys corresponding to the identity. The most efficient identity-based encryption schemes are currently based on bilinear pairings on elliptic curves such as the Weil or Tate pairings [9]. The Java Pairing Based Cryptography Library (jPBC) is a java implementation of pairing based library developed by Stanford. In proposed approach uses library routines from jPBC library for elliptic curve generation and bilinear pairing computation. The primary benefit of using elliptic curves is a smaller key size there by reduces the storage and transmission requirements.

For evaluating security in public cloud the proposed approach uses OpenStack software. OpenStack is a cloud computing platform which provides IAAS (Infrastructure as a Service). It is free and open source. In July 2010 Rackspace and NASA jointly launched an open-source cloud-software initiative. Various Linux platforms Ubuntu, Red Hat have started including OpenStack in the distribution. In 2011 Ubuntu started adapting the OpenStack and claims to provide fastest way to build cloud environment. The deployment uses Ubuntu Server 12.04 LTS with OpenStack software for inclusion of ID-PKC schemes for data encryption.

The rest of the paper is organized as follows section 3 explains security architecture of OpenStack and illustrates existing security mechanisms and section 4 discusses ID-PKC scheme in OpenStack platform. Section 5 and 6 provides security analysis of proposed security scheme and presents overview of scalability concerns when used in public cloud.

2. PREVIOUS WORKS

Some of the recent security assessments performed by Aleksandar et.al [10] revealed that OpenStack cloud software is vulnerable to attacks such as inside the cloud like “Web Server Uses Plain Text Authentication Forms” and suggests vendors to develop custom code for data transmission securely using HTTPS. The study also revealed that OpenStack is also vulnerable to attacks outside the cloud such as XSS and Web. The “Security Guidance for critical Areas of Focus in Cloud Computing” version 3.0 published in 2013 by Cloud Security Alliance states the need for using proper data encryption and securing data when in transit and stored. Using robust key management techniques for safe-guarding the keys is equally important for security.

Shamir first proposed the concept of identity based public key cryptography in 1985, in which the public key of an entity can be easily computed from his identity information. In 2001, the first practical and secure identity based encryption scheme was presented by Boneh and Franklin [9]. In Eurocrypt 2005, Sahai and Waters [13] first introduced the concept of Fuzzy Identity Based Encryption (Fuzzy IBE). In 2006, Goyal et al. [12] introduced the notion of Key-Policy Attribute-Based Encryption (KP-ABE) for fine-grained sharing of encrypted data and proposed a KP-ABE scheme that allows any monotone access structures. Bethencourt et al. [14] presented the first construction of Ciphertext policy attribute based encryption (CP-ABE). To ensure access structure requirements, in [6] there was proposed a system model using Key Policy-Attribute Based Encryption (KP-ABE) and Proxy Re-Encryption (PRE). Formally, [11] ensures data confidentiality using KP-ABE and sending the data owner delegate computation overload to the proxy using PRE.

Reference [2] provides security analysis of open source cloud for various services like compute, storage, and security and [4] provides security assessment for IaaS cloud services like virtual machines on different cloud operating systems like OpenStack, Eucalyptus. Reference [5] provides an approach to handle compromised nodes in OpenStack deployment and [8] discusses various cloud security vulnerabilities and precautions while adapting cloud for businesses.

While the ID-PKC has progressed substantially overcoming challenges with initial implementation like key-escrow, key-revocation, it was observed that ID-PKC schemes had not been evaluated in multi-tenant cloud environment like OpenStack with exception [1] provides automated audit facility for clients. This paper presents a mechanism to secure data communication in OpenStack cloud environment by modifying its security architecture and builds upon recent advancements in ID-PKC schemes.

3. SECURITY ARCHITECTURE OF OPENSTACK KEYSTONE

OpenStack comprises of various components supporting compute service (Nova), object storage (Swift), block storage (Cinder), networking (Neutron) etc. and security service called Keystone which acts as a common authentication system as shown in Fig. 3. It supports multiple authentication schemes like username and password credentials, token based and support for PKI.

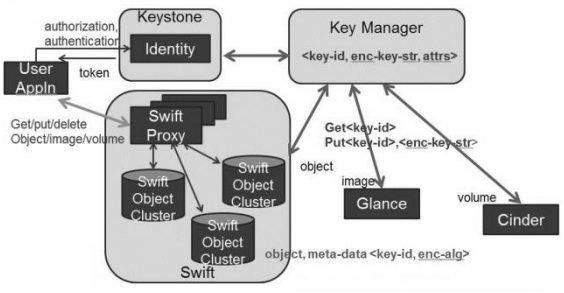

Fig 3: OpenStack keystone and key manager service providing security services to API end points

3.1 OpenStack Keystone Workflow – Token Scoping Mechanism

The keystone identity service validates user requests using UUID based tokens by maintaining map of UUID tokens to their metadata and validity. Each time when a user accesses endpoint service using UUID token the endpoint sends the UUID token to keystone which returns the token meta data like roles, tenancy. The flow is best explained by an example.

1. Alice wants to use a server. Alice gets a temporary token using the credentials. This is called unscoped token which means it is not tied to any particular tenant.
2. Using the temporary token Alice discovers list of all tenants and decides to use one particular tenant services. Alice credentials and tenant details are sent to keystone.
3. Keystone sends back tenant token also called as scoped token using which Alice can launch service using appropriate end point
4. Alice access the tenant service using new token. The tenant service authorizes the token to access the service. Service consists of set of endpoint definitions or base URLs as well as the region where the end point resides
5. The service receives the request and executes the request by creating a creating a new server and provides access details

The main drawback of this approach the single point of failure i.e., keystone service acts as a centralized server and becomes a bottleneck in presence of large number of client requests and hence the solution is not scalable to public cloud where large number of such interactions occur every moment.

3.2 PKI based token scheme in OpenStack

OpenStack is supporting PKI token scheme since Folsom release and it is default security mechanism from Grizzly release onwards. Each API endpoint holds a copy of Keystone security information like signing certificate, revocation list, CA certificate. Fig. 4 shows the major steps involved in authorizing the user using PKI based token scheme.

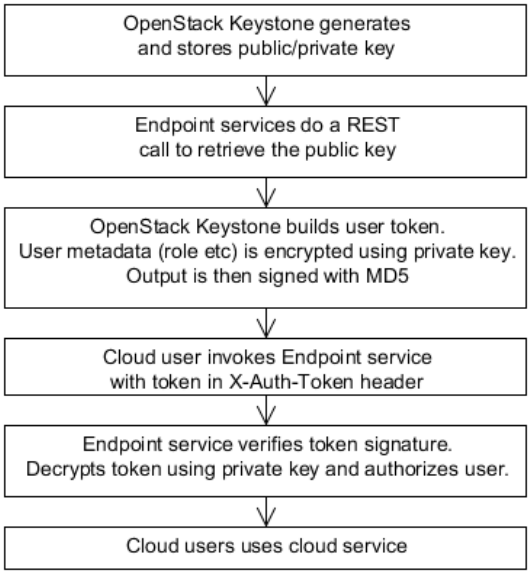

Fig 4: OpenStack keystone flowchart authorizing users using in PKI based token scoping

Keystone service generates private key, certificate and sign the same by using information like service catalog, user roles, metadata. Keystone uses delegated authorization scheme for validating user requests. All messages are encoded in "Cryptographic Message Syntax" format as per RFC 5652 guidelines. The PKI tokens enable OpenStack API endpoints to conduct offline verification of token validity by checking Keystone's signature.

The main advantage of this approach is that the auth services can accommodate large number of requests without bombarding the keystone server with requests for verification as the process of verification can be done offline.

However following problems exist with this approach

1. Each API endpoint needs to maintain a local copy of signing certificate, revocation list, CA certificate to enable auth services to its clients.
2. This data needs to be synchronized periodically with Keystone server.
3. Imagine a case where the endpoint needs to communicate to multiple security services or keystone servers. The certificate management really becomes painful activity as necessary mapping information need to be fetched from multiple keystone servers.

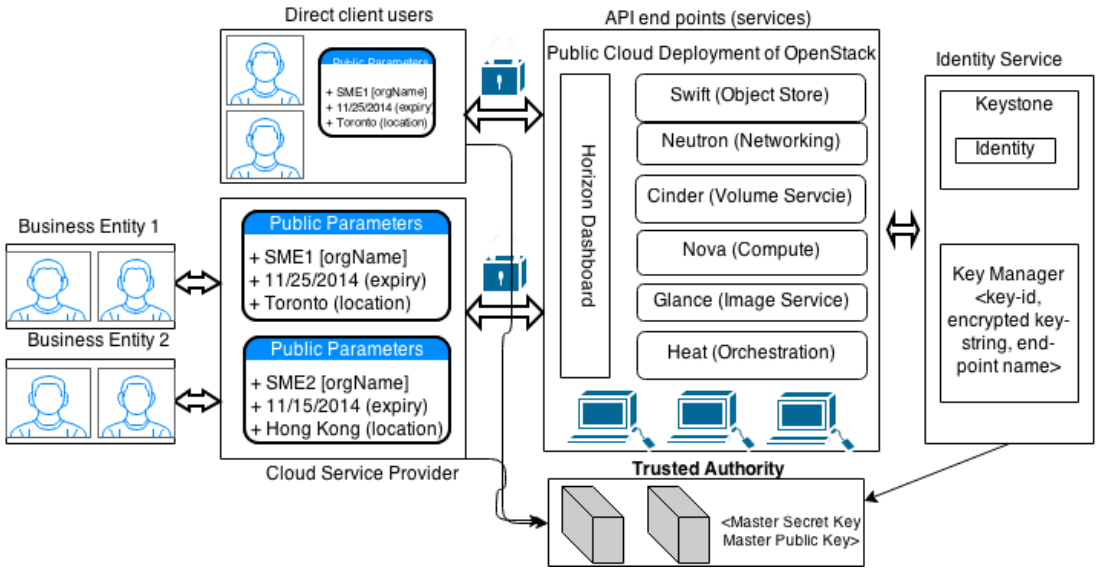

Fig 5: OpenStack platform hosted on public cloud supporting ID-PKC scheme

4. USING ID-PKC SCHEME IN CLOUD ENVIRONMENT

4.1 Identity Based Encryption Technique

Identity Based Encryption (IBE) is a public-key encryption technique that allows a public key to be calculated from an identity and a set of public mathematical parameters and that allows private key to be calculated from an identity, a set of public mathematical parameters, and a master secret key. The security of the IBE scheme is based on the assumption that the particular bilinear maps chosen are one-way functions, it is hard to calculate inverse.

The basic approach uses Weil Pairing on elliptic curve. The approach used enhanced IBE scheme which supports Key-policy attribute based encryption. Key-policy ABE is an important class of IBE scheme [11], where cipher texts are labeled with sets of attributes and private keys are associated with access structures that control which cipher texts a user is able to decrypt. Important applications include data sharing in untrusted cloud storage. The security of the scheme is based on chosen cipher text model.

4.2 KP-ABE security model used in proposed approach

4.2.1 Setup Phase

This algorithm takes as input a security parameter κ and the attribute universe U = {1, 2, . . .,N} of cardinality N. It returns the public key PK as well as a system master key.

MK as follows

PK = (Y, T1, T2, . . . , TN) (1)

MK = (y, t1, t2, . . . , tN) (2)

Where Ti ∈ G1 and ti ∈ Zp are for attribute i, 1 ≤ i ≤ N, and Y ∈ G2 is another public key component. It has Ti = gti and Y = e(g, g)y, y ∈ Zp. While PK is publicly known to all the parties in the system, MK is kept as a secret by the authority party.

4.2.2 Encryption

This algorithm takes a message M, the public key PK, and a set of attribute I as input. It outputs the ciphertext E with the following format:

E = (I, ˜ E, {Ei}i∈I) (3)

Where ˜E = MYs, Ei = Ts i , and s is randomly chosen from Zp

4.2.3 Decryption

This algorithm takes as input the ciphertext E encrypted under the attribute set I, the user's secret key SK for access tree T, and the public key PK. It first computes e(Ei, ski) = e(g, g)pi(0)s for leaf nodes. Then, it aggregates these pairing results in the bottom-up manner using the polynomial interpolation technique

4.2.4 Key Generation

This algorithm takes as input an access tree T, the master key MK, and the public key PK. It outputs a user secret key SK.

4.3 ID-PKC System Components

The ID-PKC system requires following two server side components to support security in cloud platform.

1. Public parameter Server (PPS) : a PPS server provides a well-known location for secure distribution of publicly sharable cryptographic material such as IBE public parameters, and access policy information
2. Private-key Generator (PKG): The PKG stores the master secret which is used for generating a user's IBE private key.

It system should have both PKG and PPS server per namespace, such as a DNS zone as shown in Fig. 5. This

eliminates the need for Certificate Authority used in PKI systems. In OpenStack environment the ID-PKC schemes can be used for content key encryption. Before the communication begins between clients following steps happen.

1. The PKG server generate a random Master Secret and derives public parameters from the master secret
2. Distribute Public Parameters to all clients (one time setup only)

4.4 Sending an IBE encrypted message

1. The API endpoint obtains recipient's public parameters through a secure channel like TLS 1.2 or above for security.
   IBE public parameters format:
   IBEPublicParameter ::= SEQUENCE {
   ibeAlgorithm OBJECT IDENTIFIER
   publicParameterData OCTET STRING
   }
   IBE algorithm value can be Boneh-Franklin and Boneh-Boyen depending on need.
2. In addition to this the sender requires recipient's identity, if the identity is same as identity used for communication than it is not required to request any more details. For encrypting the message, the sender chooses a content-encryption key (CEK) and uses it to encrypt the message and then encrypts the CEK with the recipient's IBE public key.

4.5 Receiving and Viewing the message

1. The recipient must have public parameters and IBE private key in order to decrypt the message. The IBE public parameters must be transported to recipients over a secure protocol.
2. The IBE private key is fetched from private key generator PKG and it is used for decrypting the CEK. The CEK is then used to decrypt the encrypted message.

4.6 Standards supporting ID-PKC schemes

1. IEEE P1636.3 - Identity-based Public-key Cryptography
2. IEEE P1363.3 - Identity Based Key Agreement Scheme
3. RFC 5408 - Identity-Based Encryption Architecture and Supporting Data Structures
4. RFC 5409 - Using the Boneh-Franklin and Boneh-Boyen Identity-Based Encryption
5. RFC 5091 - Supersingular Curve Implementations of the BF and BB1 Cryptosystems

5. DETAILED ANALYSIS OF USING ID-PKC SCHEMES IN CLOUD ENVIRONMENT

5.1 Advantages of using ID-PKC schemes in OpenStack environment

1. The public keys required for communication can be calculated by anyone who has the necessary public parameters. This eliminates the need for the sender and receiver interaction prior to sending secure messages
2. Calculation of keys (public and private) in an IBE system can occur as needed (just-in-time creation). The communicating parties my encrypt messages with no prior distribution of keys between individual participants. This is useful in cases where pre-distribution of authenticated keys is infeasible due to technical limitations.
3. A unique characteristic of IBE systems that differentiates from existing PKI schemes is that the encryption is possible without any need for communicating with server during validity period of the public parameters. This reduces network communication significantly.
4. It is possible to include role based access control policy and dynamic group information into keys and this reduces a step in security screening for authorizing users.

5.2 Problems associated in using ID-PKC schemes and risk mitigation

1. Using the Master secret key it is possible to decrypt all the communication (non-repudiation is not possible) - where in the PKG is able has access to all the communication and this PKG a high value target for adversaries. But this is desirable in certain contexts where in the organization can make out details of messages exchanged within the entity. This problem can be addressed by introducing hierarchy of private key generators instead of a single one.
2. Key revocation is a problem (since identities are used, can be overcome by using timestamp in the key) – One way to overcome this problem is by using time-stamps in generation of keys but the solution is not scalable. Goyal [13] suggests a way for overcoming the problem by using Fuzzy IBE primitives and binary tree structures.

Table 1. Summarizes security parameters for Boneh-Franklin and Boneh-Boyen algorithm with type-1 super singular curve

Security Key Size in Bits	Size of |p|	Size of |q|
80	512	160
112	1024	224
128	1536	256

5.3 Security guidelines for using ID-PKC in public cloud deployment

1. For guaranteed secure communication the public parameters defined for the system should be 80-bit, 112-bit or 128-bit encryption strength. The master secret key should not be accessible to outsiders, if it is compromised, the adversaries can recreate any user's private key and therefore decrypt all messages protected with corresponding public key.
2. The security of the system is limited by the strength of IBE key used for encryption. If the IBE key provides 112 bits of security and key used in transport uses 128-bit AES key, then security of the system is limited by the 112 bits security of IBE keys. Table 1 summarizes the security parameters for the Boneh-Franklin and Boneh-Boyan algorithms that will attain these levels of security. In this table, |p| represents the number of bits in a prime number p, and |q| represents the number of bits in a subprime q.

3. The IBE private keys should not be created using timestamp of longer duration. Due to this the problem of key compromise would be more.
4. Use ID-PKC system based on bilinear pairings such as Boneh-Franklin and Boneh-Boyen. The system is devised on family of supersingular elliptic curves over finite fields of large prime characteristic (also called as type-1 curves). For practical applications using Advanced Encryption Standard (AES) keys it is sufficient to use 128-bit levels and higher such as 192 bits or 256 bits as shown in Table I.

6. SOLUTION SCALABILITY

The performance of PKI based security model for public cloud is nearly equal to the proposed approach. However in a PKI based system each cloud service end-point has overhead of maintaining signed certificates, revocation list and CA certificates. The security information needs to be synchronized periodically with security server like keystone component and this may consume server processing power. The proposed ID-PKC system does not require certificate management as public key can be generated just-in time. So the ID-PKC systems require very less post processing during security screening of client requests as authorization policies are also part of the data, hence no separate processing for roles, access policies is required. The proposed approach scales well for multi-tenant cloud environment where volume of requests is huge and security requirements are

7. CONCLUSION

This paper explains problems associated with data security in cloud environment and issues in managing PKI certificates and trust. Security of data in cloud is still a challenge and has utmost importance as valuable customer data pertaining to enterprises is stored and accessed rapidly. A practical approach for securing data communication in cloud by using ID-PKC cryptographic techniques in OpenStack cloud environment is presented in this paper. Identity based cryptographic system is certificate free, well suited for providing the security in cloud world. The new system is less vulnerable to spam and it also enables postdating of messages for future decryption. It is much efficient in terms of performance analysis. It also enables automatic expiration, rendering messages unreadable after a certain date. Implementing IBE system in cloud requires far simpler infrastructure (meaning fewer servers and easier installation) then public key infrastructure security system. The paper presents do's and don'ts for implementing ID-PKC based solution for OpenStack cloud environment and provided security analysis.

8. REFERENCES

[1] Ullah K.W, Ahmed A.S. and Ylitalo J, "Towards Building an Automated Security Compliance Tool for the Cloud", IEEE Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 1587-1593, July 2013

[2] Ristov S, Gusev M, "Security evaluation of open source clouds", IEEE EUROCON 2013, pp. 73-80, July 2013

[3] Fakhar F, Shibli M.A, "Management of Symmetric Cryptographic Keys in cloud based environment", IEEE Advanced Communication Technology (ICACT) 2013, pp. 39-44, Jan 2013

[4] Donevski A, Ristov S, Gusev M, "Security assessment of virtual machines in open source clouds", IEEE Information & Communication Technology Electronics & Microelectronics (MIPRO), pp. 1094-1099, May 2013

[5] Taheri Monfared A, Jaatun M.G, "As Strong as the Weakest Link: Handling Compromised Components in OpenStack", IEEE Cloud Computing Technology and Science (CloudCom), pp. 189-196, Dec. 2011

[6] Y. Chen, V. Paxson, and R. H. Katz, "What's New About Cloud Computing Security?" EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2010-5, Jan. 2010

[7] Hongwei Li, Yuanshun Dai, Bo Yang, "Identity-Based Cryptography for Cloud Security", IACR Cryptology, Jan 2011

[8] Ashish Kumar, "World of Cloud Computing & Security", Vol.1, No.2, International Journal of Cloud Computing and Services Science (IJ-CLOSER), pp. 53-58, Jun 2012

[9] D. Boneh and M. K. Franklin, "Identity-based encryption from the Weil pairing", CRYPTO, LNCS 2139, Springer-Verlag, pp. 213–229, 2001

[10] Sasko Ristov, Marjan Gusev and Aleksandar Donevski, "OpenStack Cloud Security Vulnerabilities from Inside and Outside", Cloud Computing, The Fourth International Conference on Cloud Computing, GRIDs, and Virtualization, 2013, pp. 95-101, 2013

[11] Chang-Ji Wang and Jian-Fa Luo "A Key-policy Attribute-based Encryption Scheme with Constant Size Ciphertext", Eighth International Conference on Computational Intelligence and Security, pp. 447-451, Nov 2012

[12] Sahai and B. Waters, "Fuzzy Identity Based Encryption", In EUROCRYPT 2005, LNCS 3494, Springer-Verlag, 2005

[13] V. Goyal, O. Pandey, A. Sahai and B. Waters, "Attribute Based Encryption for Fine-Grained Access Conrol of Encrypted Data‖", ACM conference on Computer and Communications Security, 2006

[14] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute based encryption" Proc. of IEEE Symposium on S&P, 2007.