IPHV7I3016X
International Journal Of Advanced Research and Innovation -Vol.7, Issue .III
ISSN Online: 2319 – 9253
Print: 2319 – 9245

EVOLUTION OF HEALTH CARE MONTORING SYSTEM IN CLOUD COMPUTING ENVIRONMENT USING AES ALGORITHAM

#1P.Havilah -M.Tech Pursuing,
#2B.Rajani –Associate Professor,
Department of Computer Science & Engineering,
CMR COLLEGE OF ENGINEERING AND TECHNOLOGY, Hyd, TS, India.

Abstract: - content sharing environments such as social networking are very dynamic in terms of the number of on-line users, storage requirement, network bandwidth, computational capability, applications and platforms, thus it is not easy for a service provider to allocate resources following the traditional client-server model. In this paper, we circumvent these obstacles and close this gap by proposing a novel key management paradigm. The new paradigm is a hybrid of traditional broadcast encryption and group key agreement. In proposed System, proposed an information management architecture using CP-ABE and optimized security enforcement efficiency. Furthermore, they employed the architecture and optimization method on two example applications: An HIPAA (Health Insurance Portability and Accountability Act) compliant distributed file system and a content delivery network. An approach to access control in content sharing services is to empower users to enforce access controls on their data directly, rather than through a central administrator. However, this requires flexible and scalable cryptographic key management to support complex access control policies. A native access control solution is to assign one key for each user attribute, distribute the appropriate keys to users who have the corresponding attributes, and encrypt the media with the attribute keys repeatedly.

In such a system, each member maintains a single public/secret key pair. Upon seeing the public keys of the members, a remote sender can securely broadcast to any intended subgroup chosen in an *ad hoc* way Multi-message Ciphertext Policy Attribute-Based Encryption (MCP-ABE) technique, and employs the MCP-ABE to design an access control scheme for sharing scalable media based on data consumers' attributes (e.g., age, nationality, gender) rather than an explicit list of the consumers' names. The scheme is efficient and flexible because MCP-ABE allows a content provider to specify an access policy and encrypt multiple messages within one cipher text such that only the users whose attributes satisfy the access policy can decrypt the cipher text.

Keywords: - *Broadcast, Access Control, Key Management, Data Security.*

I.INTRODUCTION

An attribute-based access control (ABAC) system is a strategy for making runtime decisions about what features or data a user can access in an application, based on a combination of policies and data about both the user and transaction context.

Data about the user typically comes in the form of identity attributes -- things like the user's name, login ID, department, location, job role, etc. This data normally comes from an LDAP directory.

Data about transaction context includes what operation the user is attempting to perform, what data the user would access through this operation, the current time and date, the location of the user (e.g., IP address or similar), the type of device from which the user connected (e.g., web user agent or similar) and how the user authenticated.

Policy data links operations and data to identity and transaction data, to make runtime go/no-go decisions. There is an XML standard for expressing such policy decisions, called XACML. XACML stands for extensible Access Control Markup Language.

Access control is the fundamental security mechanism to facilitate information sharing in a controllable manner. It exerts control over which user can access which resource based on a permission relationship between user attributes and resource attributes, where attributes can be any information deemed relevant for granting access, such as user's job function and resource quality, and permission is specified in terms of requirements on the attributes of resource and user. Any user with attributes that meet the requirements has access to that resource. However, it is challenging to design a suitable access control mechanism in content sharing services due to: (1) any individual is able to freely produce any number and any kind of online media such as text, image, sound, video, and presentation; (2) any individual is able to grand any access to his media to anyone, at any time; (3) an individual may reveal a large number of attributes (*e.g.* name, age, address, friendship, classmate, fans, hobby, personal interest, gender, and mobility), and some of them can be very dynamics; and (4) individuals may share contents using various devices and bandwidth, and hence demand different access privileges for the same media.

In this paper we present an access control scheme for scalable media. The scheme has several benefits which make it especially suitable for content delivery. For example, it is extremely scalable by allowing a data owner to grant data access privileges based on the data consumers' attributes (*e.g.*, age, nationality, gender) rather than an explicit list of user names; and it ensures data privacy and exclusiveness of access of scalable media by employing attribute-based encryption. For this purpose, we introduce a novel Multi-message Ciphertext Policy Attribute-Based Encryption (MCP-ABE) technique. MCP-ABE encrypts multiple messages within one ciphertext so as to enforce flexible attribute-based access control on scalable media. Specifically, the scheme constructs a key graph which matches users' access privileges, encrypts media units with the corresponding keys, and then encrypts the key graph with MCP-ABE; only those data consumers with the required user attributes can decrypt the encryption of the key (sub)graph and then decrypt the encrypted media units. To cater for resource-limited mobile devices, the scheme offloads computational intensive operations to cloud servers while without compromising user data privacy. Attribute-based encryption schemes, such as CP-ABE and MCP-ABE, are designed to be secure against user collusion attacks. The present scheme is also secure against user collusion attacks due to use of attribute-based encryption. The experiments demonstrate that the present scheme is applicable on smartphone, especially when a cloud platform is available.

II. RELATED WORK

Cloud computing promises lower costs, rapid scaling, easier maintenance, and service availability anywhere, anytime, a key challenge is how to ensure and build confidence that the cloud can handle user data securely. A recent Microsoft survey found that "58 percent of the public and 86 percent of business leaders are excited about the possibilities of cloud computing. But more than 90 percent of them are worried about security, availability, and privacy of their data as it rests in the cloud."

Cloud computing is the provision of dynamically scalable and often virtualized resources as a services over the internet Users need not have knowledge of, expertise in, or control over the technology infrastructure in the "cloud" that supports them. Cloud computing represents a major change in how we store information and run applications. Instead of hosting apps and data on an individual desktop computer, everything is hosted in the "cloud"—an assemblage of computers and servers accessed via the Internet.

❖ Cloud computing exhibits the following key characteristics:

1. **Agility** improves with users' ability to re-provision technological infrastructure resources.

2. **Multi tenancy** enables sharing of resources and costs across a large pool of users thus allowing for:

3. **Utilization &Efficiency** improvements for systems that are often only 10–20% utilized.

4. **Reliability** is improved if multiple redundant sites are used, which makes well-designed cloud computing suitable for business continuity and disaster recovery.

5. **Performance** is monitored and consistent and loosely coupled architectures are constructed using web services as the system interface.

6. **Security** could improve due to centralization of data, increased security-focused resources, etc., but concerns can persist about loss of control over certain sensitive data, and the lack of security for stored kernels. Security is often as good as or better than other traditional systems, in part because providers are able to devote resources to solving security issues that many customers cannot afford. However, the complexity of security is greatly increased when data is distributed over a wider area or greater number of devices and in multi-tenant systems that are being shared by unrelated users. In addition, user access to security audit logs may be difficult or impossible. Private cloud installations are in part motivated by users' desire to retain control over the infrastructure and avoid losing control of information security.

7. **Maintenance** of cloud computing applications is easier, because they do not need to be installed on each user's computer and can be accessed from different places.

The major security concern in group-oriented communications with access control is key management. Existing key management systems in these scenarios are mainly implemented with two approaches referred to as group key agreement (or group key exchange by some authors) and key distribution systems (or the more powerful notion of broadcast encryption). Both are active research areas having generated large respective bodies of literature. Group key agreement allows a group of users to negotiate a common secret key via open insecure networks. Then, any member can encrypt any confidential message with the shared secret key and only the group members can decrypt. In this way, a confidential intra group broadcast channel can be established without relying on a centralized key server to generate and distribute secret keys to the potential members. A large number of group key agreement protocols have been proposed. The earlier efforts focused on efficient establishment of the initial group key. If more than this threshold of users is revoked, the scheme will be insecure and hence not fully collusion-resistant. Subsequently, by exploiting newly developed bilinear paring technologies, a fully collusion-resistant public-key broadcast encryption scheme was presented that has complexity in key size, ciphertext size, and computation cost, where the maximum allowable number is of potential receivers. A recent scheme reduces the size of the key and the ciphertexts, although it has the same asymptotical sublinear complexity. An up-to-date scheme was presented in, which strengthens the security concept of public-key broadcast encryption schemes while keeping the same complexity.

EXISTING METHOD:

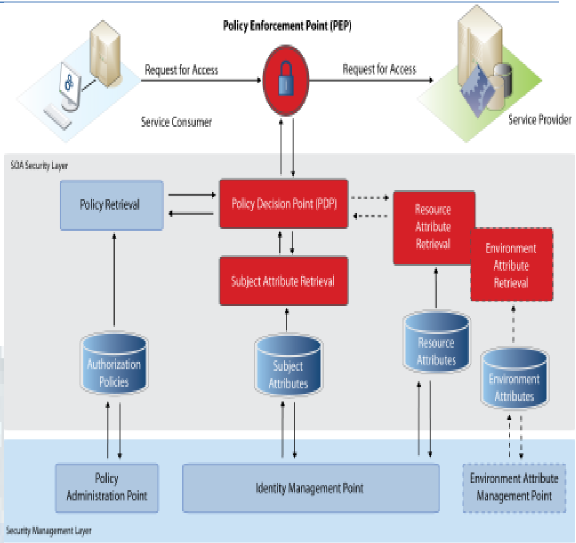
In Existing System, there are multiple owners who may encrypt according to their own ways, possibly using different sets of cryptographic keys. Letting each user obtain keys from every owner who's records she wants to read would limit the accessibility since patients are not always online. An alternative is to employ a central authority (CA) to do the key management on behalf of all record owners, but this requires too much trust on a single authority (i.e., cause the key escrow problem). Key escrow (also known as a "fair" cryptosystem) is an arrangement in which the keys needed to decrypt encrypted data are held in escrow so that, under certain circumstances, an authorized third party may gain access to those keys. These third parties may include businesses, who may want access to employees' private communications, or governments, who may wish to be able to view the contents of encrypted communications.

PROPOSED METHOD:

In this paper we present an access control scheme for scalable media. The scheme has several benefits which make it especially suitable for content delivery. For example, it is extremely scalable by allowing a data owner to grant data access privileges based on the data consumers' attributes (e.g., age, nationality, gender) rather than an explicit list of user names; and it ensures data privacy and exclusiveness of access of scalable media by employing attribute-based encryption. For this purpose, we introduce a novel Multi-message Ciphertext Policy AttributeBased Encryption (MCP-ABE) technique. MCP-ABE encrypts multiple messages within one ciphertext so as to enforceflexible attribute-based access control on scalable media. Specifically, the scheme constructs a key graph which matches users' access privileges, encrypts media units with the corresponding keys, and then encrypts the key graph with MCP-ABE; only those data consumers with the required user attributes can decrypt the encryption of the key (sub) graph and then decrypt the encrypted media units. To cater for resource-limited mobile devices, the scheme offloads computational intensive operations to cloud servers while without compromising user data privacy.

III. IMPLEMENTATION

Key to any successful ABAC implementation will be the Attribute Services (AS). AS makes attribute collection, dissemination, and security possible through the use of Policy Decision Points (PDPs) and Policy Enforcement Points (PEPs). For example, when a user tries to access a resource, the PEP will defer to the PDP, whose job it is to decide whether or not to authorize the user based on the description of the user's attributes. Policies stored on the PEP and PDP systems provide the rule sets around which decisions are made.


Policy Enforcement Point (PEP)
Request for Access
Request for Access
Service Consumer
Service Provider
SOA Security Layer
Policy Retrieval
Policy Decision Point (PDP)
Resource Attribute Retrieval
Environment Attribute Retrieval
Subject Attribute Retrieval
Authorization Policies
Subject Attributes
Resource Attributes
Environment Attributes
Policy Administration Point
Identity Management Point
Environment Attribute Management Point
Security Management Layer

Typically, however, variations exist in attribute data due to a lack of process, training, typographical errors, etc. (i.e., officer rank may be encoded as "LtCol"or "Lieutenant Colonel"). This problem grows exponentially with the number of attributes and interconnected agencies, departments and systems. Additionally, making attribute data available externally can pose a significant security risk.

Disk encryption is a technology which protects information by converting it into unreadable code that cannot be deciphered easily by unauthorized people. Disk encryption uses disk encryption software or hardware to encrypt every bit of data that goes on a disk or disk volume. Disk encryption prevents unauthorized access to data storage. The term "full disk encryption" (or whole disk encryption) is often used to signify that everything on a disk is encrypted, including the programs that can encrypt bootable operating system partitions. But they must still leave the master boot record (MBR), and thus part of the disk, unencrypted. There are, however, hardware-based full disk encryption systems that can truly encrypt the entire boot disk, including the MBR.

Our work focuses on an important class of widely used applications that includes e-mail, personal financial management, social networks, and business tools such as word processors and spreadsheets. The following criteria define this class of applications:

- ➢ Provide services to a large number of distinct end users, as opposed to bulk data processing or workflow management for a single entity;
- ➢ Use a data model consisting mostly of sharable units, where all data objects have access control lists (ACLs) with one or more users; and
- ➢ Developers could run the applications on a separate computing platform that encompasses the physical infrastructure, job scheduling, user authentication, and the base software environment, rather than implementing the platform themselves.

1).Key Policy Attribute-Based Encryption (KP-ABE):

KP-ABE is a public key cryptography primitive for one-to-many communications. In KP-ABE, data are associated with attributes for each of which a public key component is defined. User secret key is defined to reflect the access structure so that the user is able to decrypt a cipher text if and only if the data attributes satisfy his access structure. A KP-ABE scheme is composed of four algorithms which can be defined as follows:

- Setup Attributes
- Encryption
- Secret key generation
- Decryption

Setup Attributes:

This algorithm is used to set attributes for users. From these attributes public key and master key for each user can be determined. The attributes, public key and master key are denoted as

Attributes- $U = \{1, 2. . . N\}$
Public key- $PK = (Y, T1, T2, . . . , TN)$
Master key- $MK = (y, t1, t2, . . . , tN)$

Encryption:

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

$$E = (I, \tilde{}\ E, \{Ei\}i\ )$$

where $\tilde{}E = MY, Ei = Ti.$

Secret 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* as follows.

$$SK = \{ski\}$$

Decryption:

This algorithm takes as input the cipher text ***E*** encrypted under the attribute set **U**, the user's secret key ***SK*** for access tree ***T***, and the public key ***PK***.

Finally it output the message ***M*** if and only if **U satisfies T**.

2) Proxy Re-Encryption (PRE):

Proxy Re-Encryption (PRE) is a cryptographic primitive in which a semi-trusted proxy is able to convert a cipher text encrypted under Alice's public key into another cipher text that can be opened by Bob's private key without seeing the underlying plaintext. A PRE scheme allows the proxy, given the proxy re-encryption key

$$rka\leftrightarrow b,$$

to translate cipher texts under public key ***pk1*** into cipher texts under public key ***pk2*** and vise versa.

3) Lazy re-encryption:

The lazy re-encryption technique and allow Cloud Servers to aggregate computation tasks of multiple operations. The operations such as

- Update secret keys
- Update user attributes.

IV. CP-ABE BASED SECURED CLOUD STORAGE ARCHITECTURE

In this section, first, we give a formal definition of our proposed scheme, and later we give the security model in which our scheme is proven to be secure.

4.1. System Description

Fig. 1 shows the architecture of the data sharing system, which consists of the following system entities:

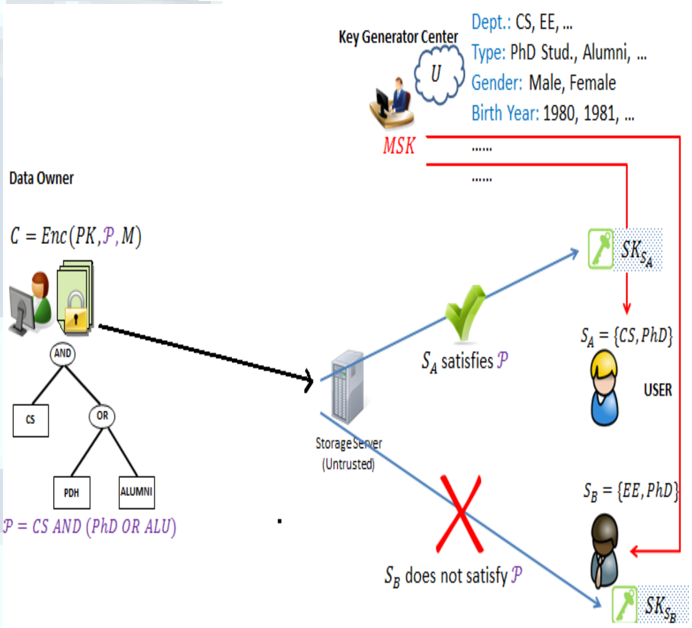


Figure 1. CP-ABE based Storage Architecture

Key generation center: It is a key authority that generates public and secret parameters for CP-ABE. It is in charge of issuing, revoking, and updating attribute keys for users. It grants differential access rights to individual users based on their attributes. It is assumed to be honest but curious. That is, it will honestly execute the assigned tasks in the system; however, it would like to learn information of encrypted contents as much as possible. Thus, it should be prevented from accessing the plaintext of the encrypted data even if it is honest.

Data storing center: It is an entity that provides a data sharing service. It is in charge of controlling the accesses from outside users to the storing data and providing corresponding contents services. The data storing center is another key authority that generates personalized user key with the KGC, and issues and revokes attribute group keys to valid users per each attribute, which are used to enforce a fine-grained user access control. Similar to the previous schemes [2],[3],[4], we assume the data storing center is also semi-trusted (that is, honest-but-curious) like the KGC.

Data owner: It is a client who owns data, and wishes to upload it into the external data storing center for ease of sharing or for cost saving. A data owner is responsible for defining (attribute- based) access policy, and enforcing it on its own data by encrypting the data under the policy before distributing it.

User: It is an entity who wants to access the data. If a user possesses a set of attributes satisfying the access policy of the encrypted data, and is not revoked in any of the valid attribute groups, then he will be able to decrypt the ciphertext and obtain the data.

4.2.Ciphertext-Policy Attribute-Based Encryption and Access Format

In our construction private keys will be identified with a set of descriptive attributes. A party that wishes to encrypt a message will specify through an access tree structure a policy that private keys must satisfy in order to decrypt. Each interior node of the tree is a threshold gate and the leaves are associated with attributes. A user will be able to decrypt a ciphertext with a given key if and only if there is an assignment of attributes from the private key to nodes of the tree such that the tree is satisfied. We use the same notation as[5] to describe the access trees ,even though in our case the attribute s are used to identify the keys(as opposed to the data).specified in the private key, while the ciphertexts are simply labeled with a set of descriptive.

Access tree T structure : Let T be a tree representing an access structure. Each non-leaf node of the tree represents a threshold gate, described by its children and a threshold value. If num x is the number of children of a node x and k, x is its threshold value, then 0 <kx = numx. When kx = 1, the threshold gate is an OR gate and when kx = numx, it is an AND gate. Each leaf node x of the tree is described by an attribute and a threshold value kx = 1.

To facilitate working with the access trees, we define a few functions. We denote the parent of the node x in the tree by parent(x). The function att(x) is defined only if x is a leaf node and denotes the attribute associated with the leaf node x in the tree. The access tree T also defines an ordering between the children of every node, that is, the children of a node are numbered from 1 to num. The function index(x) returns such a number associated with the node x.Where the index values are uniquely assigned to nodes in the access structure for a given key in an arbitrary manner.

Satisfying an access tree : Let T be an access tree with root r. Denote by Tx the sub tree of T rooted at the node x. Hence T is the same as Tr. If a set of attributes Y satisfies the access tree Tx, we denote it as Tx(Y) = 1. We compute Tx(Y) recursively as follows. If x is a non-leaf node, evaluate Tx (Y) for all children x of node x. Tx(Y) returns 1 if and only if at least kx children return 1. If x is a leaf node, then Tx(Y) returns 1 if and only if att(x) Є Y.

Encrypt(PK,M,T) : The encryption algorithm encrypts a message M under the tree access structure T . The algorithm first chooses a polynomial qx for each node x (including the leaves) in the tree T . These polynomials are chosen in the following way in a top-down manner, starting from the root node R. For each node x in the tree, set the degree dx of the polynomial qx to be one less than the threshold value kx of that node, that is,dx = kx -1. Starting with the root node R the algorithm chooses a random s ? p and sets qR(0) = s. Then, it chooses Z dR other points of the polynomial qR randomly to define it completely. For any other node x, it sets qx(0) =qparent(x)(index(x)) and chooses dx other points randomly to completely define qx. Let, Y be the set of leaf nodes in T . The ciphertext is then constructed by giving the tree access structure T and computing

$$\mathbf{CT = (T , C = Me(g,g)^{\alpha s} ,\ C = h^{s} ,}$$

$$\mathbf{\forall y\in Y : Cy = g^{qy} ,Cy = H(att(y))^{qy(0)}).}$$

KeyGen(MK,S). The key generation algorithm will take as input a set of attributes S and output a key that identifies with that set The algorithm first chooses a random r ? ,Z p and then random r ? Z for each attribute j ? S. Then it j p computes the key

$$\mathbf{SK = (D = g^{(\alpha+r)/\beta},}$$
$$\mathbf{\forall j \in S : Dj= g^{r}\cdot H(j)^{rj}, Dj= g^{rj}).}$$

Delegate(SK,S). The delegation algorithm takes in a secret key SK, which is for a set S of attributes, and another set S^1 such that S^1 Є S. The secret key is of the form SK = (D, ∀j ∈ S : Dj,D^1j). Then algorithm chooses random the r˜and r ˜k ∀ k Є S`. Then it creates a new secret key as

$$\mathbf{SK^{\sim}=(D^{\sim}= Df^{r}, \forall k\in S^{\sim}: Dk= Dkg^{r}H(k)^{rk},D^{\sim}k= D^{-}kg^{rk})}$$

The resulting secret key SK is a secret key for the set S. Since the algorithm re-randomizes the key, delegated key is equivalent to one received directly from the authority.

V.CONCLUSION

We proposed, in this work, is mandatory to provide the security to data in personal/private involves in online nature. Some of the data centers are utilizes the high-end protection systems to provide the security to the data. we know that the data centers are nothing but a cloud adding protection to the single cloud it leads to all services provided by client meansthousands o services are benefited and TBs of client data are protected.

REFERENCES

[1] H. Abu-Libdeh, L. Princehouse and H. Weatherspoon, "RACS: a case for cloud storage diversity", SoCC'10:Proc. 1st ACM symposium on Cloud computing, 2010, pp. 229-240.

[2] P. Ammann and S. Jajodia, “Distributed Timestamp Generation in Planar Lattice Networks,” ACM Trans. Computer Systems, vol. 11, pp. 205-225, Aug. 1993.

[3] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable Data Possession at Untrusted

[4] Stores,” Proc. ACM Conf. Computer and Comm. Security, pp. 598-609, 2007.

[5] E. Barka and A. Lakas, “Integrating Usage Control with SIP-Based Communications,” J. Computer Systems, Networks, and Comm.,vol. 2008, pp. 1-8, 2008.

[6] D. Boneh and M.K. Franklin, “Identity-Based Encryption from the Weil Pairing,” Proc. Int’l Cryptology Conf. Advances in Cryptology, pp. 213-229, 2001.

[7] R. Bose and J. Frew, “Lineage Retrieval for Scientific Data Processing: A Survey,” ACM Computing Surveys, vol. 37, pp. 1-28, Mar. 2005.

[8] P. Buneman, A. Chapman, and J. Cheney, “Provenance Management in Curated Databases,” Proc. ACM SIGMOD Int’l Conf.

[9] Management of Data (SIGMOD ’06), pp. 539-550, 2006.

[10] B. Chun and A.C. Bavier, “Decentralized Trust Management and Accountability in Federated Systems,” Proc. Ann. Hawaii Int’l Conf. System Sciences (HICSS), 2004.

[11] C. Dwork, “The Differential Privacy Frontier Extended Abstract,” *Proc. 6th Theory of Cryptography Conf.* (TCC 09),LNCS 5444, Springer, 2009, pp. 496-502.

[12] C. Gentry, “Fully Homomorphic Encryption Using Ideal Lattices,” *Proc. 41st Ann. ACM Symp. Theory Computing* (STOC 09), ACM, 2009, pp. 169-178.

[13] E. Naone, “The Slow-Motion Internet,” *Technology Rev.*, Mar./Apr. 2011; www.technologyreview.com/files/54902/

[14] GoogleSpeed_charts.pdf.Greenberg, “IBM’s Blindfolded Calculator,”*Forbes*,13 July 2009; www.forbes.com/forbes/2009/0713/breakthroughs-privacy-super-secret-encryption.html.

[15] Cloud Data Protection for the Masses , Dawn Song, Elaine Shi, and Ian Fischer, *University of California, Berkeley* Umesh Shankar, *Google*, 2012 IEEE Published by the IEEE Computer Society JANUARY 2012

[16] Abraham, G. Chockler, I. Keidar and D. Malkhi, "Byzantine disk paxos: optimal resilience with Byzantine shared memory", Distributed Computing, 18(5), 2006, pp. 387-408.