# An Enhancement of Cloud Data Access Security using Identity based Encryption

**Pallavi K P[1] Girish[2] Dr. H. D. Phaneendra[3]**
[1]M. Tech-CNE [2]Associate Professor [3]Professor
[1,2]Department of PGSCEA [3]Department of Computer Science & Engineering
[1,2,3]The National Institute of Engineering, Mysore, India

*Abstract*— Identity based Secure distributed data storage is a scheme that reduces the burden of maintaining excessive number of files from the owner to proxy servers. The Proxy servers are used to convert encrypted files for the owner to encrypted files for the receiver without the necessity of knowing the content of the original files. In practice, the owner removes the original files for the sake of space efficiency. Hence, the issues of confidentiality and integrity of the outsourced data must be addressed carefully since the cloud is managed by untrusted third party. In this paper, we propose an identity-based secure distributed data storage (IBSDDS) schemes. Our schemes captures the following properties: (1) The owner of the file can decide the access permission independently without the help of the private key generator (PKG); (2) For one request, a receiver can access only one file, instead of all the files of the owner; (3) Our scheme is secure against the collusion attacks and untrusted users, namely even if the receiver can compromise the proxy servers, it's not possible for him to obtain the owner's secret key. Although this system is secure against different types of attack, by using the concept of re-encryption the data gets more secured and the access permission that who will access the data is decided by the owner himself. In existing system to provide better security data owner has to be online all the time so our propose system will be helpful for data owner by getting notification about the request of the user to their mail-ids.

***Key words:*** Distributed Data Storage, Identity, Private- Key Generator, Access Permission

## I. Introduction

Cloud computing provides users with a convenient mechanism to manage their personal files with the concept called database-as-a-service (DAS). In DAS schemes, a user can outsource his encrypted files to untrusted proxy servers. A proxy server performs some functions on the outsourced cipher texts without knowing anything about the content of the original files/data. Unfortunately, this technique has not been employed extensively. The cloud server is untrusted server because it is managed by an untrusted third party. Therefore, the issues of confidentiality comes into existence where data owner is mostly concerned on the security from unauthorized access, integrity means correctness of the file/data after outsourcing to the proxy server, as well as the data should not be modified by unauthorized user or even through by the proxy server. So, this is the reason that is becoming major research problem among research community and it growing day by day.

After outsourcing the files to proxy servers, the user will remove those files from his local machine. Therefore, how to guarantee that outsourced files are not accessed by the unauthorized users and not modified by proxy servers is an important problem that has been considered in the data storage research community. Likewise, how to guarantee that an authorized user can access the outsourced files from proxy servers is another concern as the proxy server only maintains the outsourced cipher texts.

Confidentiality is proposed to prevent unauthorized users from accessing the sensitive data as it is subject to unauthorized disclose and access after being outsourced. Since the introduction of DAS, the confidentiality of outsourced data has been the primary focus among the research community. To provide confidentiality to the outsourced data, the encryption schemes are deployed. Integrity can prevent outsourced data from being replaced and modified. Some of the schemes have been proposed to protect the integrity of the outsourced data, such as proof of retrievability and provable data possession. These schemes, digital signature schemes and message authentication codes (MAC) are deployed. Query in data storage is executed between a receiver and a proxy server. The proxy server can perform some functions on the outsourced cipher texts and convert them to those for the receiver. As a result, the receiver can obtain the data outsourced by the owner without the Proxy server knowing the content of the data.

## II. Cloud Computing

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name is taken from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud Computing authorize remote services with a user's data, software and also computation. Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services often provide access to advanced software applications and high-end networks of server computers. The structure of cloud computing is shown in figure 1.

Cloud computing relies on sharing of resources to achieve coherence and economies of scale, similar to a utility (like the electricity grid) over a network. At the foundation, the cloud computing is the broader concept of converged infrastructure and shared services. Cloud computing, or in a simple way the shorthand just "the cloud", also focuses on maximizing the effectiveness of the shared resources. Cloud computing resources are not only shared by multiple users but they are also dynamically reallocated for each demand by the user. This idea can work for allocating resources to users. For example, a cloud computing facility that serves European users during European business hours with a specific application (e.g., email) may reallocate the same resources to serve North American users during North America's business hours with a different application (e.g., a web server). This approach should maximize the use of computing power thus reducing environmental damage as well since less power; rack space, air conditioning, and many more are required for a variety of functions to be done. With the use of cloud computing, multiple users can access one server to retrieve and update their required data without purchasing licenses for different applications.

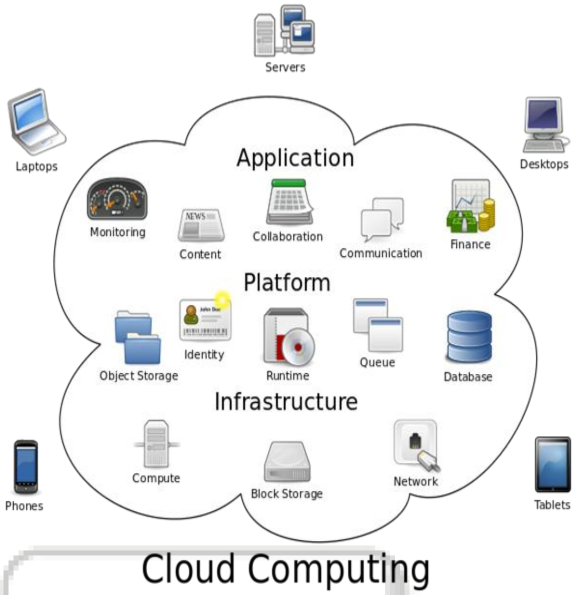

Fig. 1: Structure of Cloud Computing

Cloud computing technology includes many technologies such as the autonomic computing virtualization, service oriented architecture, utility computing and many more. The purpose of these technologies is to provide scalable, shared resources (software and hardware) and providing services over the network. The cloud term 'as a service' is referred to as providing something as a service over the network. In cloud computing, users can utilize powerful computing resources and obtain ample storage spaces. This is called Database-as-a-Service (DaaS), Software-as-a-Service (SaaS) or Infrastructure as a Service (IaaS) and many more. All the services privided are based on policy of on-demand fashion in which users can pay only to for their required usage. Today, many cloud service providers such as Amazon Web Services, Microsoft's Windows Azure and Office 365, Oracle cloud etc. are providing the facility to different users. Users who cannot afford huge cost to build their own huge infrastructure, they can have their work done by the help of cloud providers at lower cost.

As per the type of users and the hosting of environment the cloud architecture can be divided as four types: Public Cloud, Private Cloud, Hybrid Cloud, and Community Cloud. Public cloud provides services, which are hosted for public usage and anybody can have their data stored and get done the services using the public cloud. Data security is most important issue here. In Private cloud, the data access and service usage are restricted to single authority only. In Hybrid cloud, it is shared by a limited no. of organizations and it combines the features of both private and public cloud. Community cloud is similar to private cloud but in here the data is shared among the same entities of the one organization.

Cloud computing provides a service called as database-as-a-service (DaaS), here the data owner can outsource his data/files on cloud server for reducing space cost as well as maintenance cost and only the authorized users can query/request this data. In this before uploading data on cloud server data owner has to be encrypt his data. Proxy servers can perform some functions on the stored cipher text without knowing anything about the original data/files.

## III. Identity Based Secure Distributed Data Storage Scheme (IBSDDS)

In identity-based secure distributed data storage (IBSDDS) scheme, a user's identity can be an arbitrary string and two parties can communicate with each other without checking the public key certificates. At first, the owner encrypts his files under his identity prior to outsourcing them to proxy servers. Later owner sends the cipher texts to the proxy servers. The proxy servers can transfer a cipher text encrypted under the identity of the owner to a cipher text encrypted under the identity of the receiver after they obtain access permission (re encryption key) from the owner. To provide confidentiality for the outsource data, an efficient IBSDDS scheme should provide the following properties.

### A. *Unidirectional:*

After receiving the access permission (re-generation key) from Alice, the proxy server can transfer a cipher text for Alice to a cipher text for Bob while he cannot transfer a cipher text for Bob to a cipher text for Alice.

### B. *Non-Interactive:*

Access permission can be decided by the file owner without any trusted third party and interaction with him.

### C. *Key Optimal:*

The secret key size of the receiver is constant and independent of the delegations which he accepts.

### D. *Collusion-Safe:*

The file owner's secret key is secure even if the receiver can compromise the proxy server.

### E. *Non-Transitive:*

By receiving the access permissions computed by Alice for Bob and Bob for Charlie, the proxy server cannot transfer a cipher text for Alice to a cipher text for Charlie.

### F. *File-Based Access:*

The receiver can access only one file, for one request. This can improve the security of the outsourced files and is desirable to maintain the access record.

## IV. Proposed Framework

In this paper, we propose a scheme of identity-based secure distributed data storage (IBSDDS) schemes where, the receiver can access only one file of the owner, instead of all files, for one request. Alternatively, access permission (re-encryption key) is constrained not only to the identity of the receiver but also for the file. The access permission will be decided by the owner, instead of the trusted third party (PKG). Noticeably, our scheme is secure against the collusion attacks. When re-generation key is generated by owner for the receiver's request for a file, it is sent to receiver's mail ID. Receiver can login to his mail id and access authentication key (re-generation key). To enhance the security level we provide re-generation key to receiver's mail-id.

### A. *System Architecture of Proposed System:*

The distributed data storage schemes analysis task is going to establish complete information about the concept, behavior and the other constraints like performance measure and the system optimization. The main goal of Distributed Data Storage Schemes Analysis is to completely specify the technical details for the main concept in a concise and unambiguous manner. Architectural Diagram Analysis is the process of understanding the environment in which a proposed system or systems will operate and determining the requirements for the system. The system architecture of the proposed system is shown in fig 2.

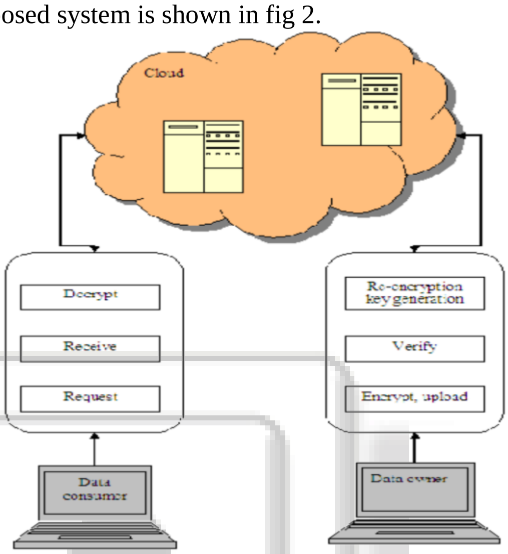

Fig.2: System Architecture Of Distributed Storage System In Cloud Environment

#### 1) *Advantages of Proposed system:*

In proposed system the user can upload the data and download the required data. Here both sender and receiver can upload and access the data by registering their names with their ID's. These ID's are stored in the proxy servers and are invisible.

1) The user can act as both data owner and data consumer.
2) Any user can get the required data by registering with them.
3) The proposed system provides more confidentiality.
4) The proposed work provides the facility to system users to access the system and can get the request notification to their mail-ids.

### B. *Algorithms and Techniques used:*

We need so many algorithms, the basic and important algorithms among all the algorithms which we are using is described below:

#### 1) *Setup Algorithm:*

The setup algorithm takes as input a security parameter 1_, and outputs the public parameters and a master secret MSK.

#### 2) *Key-Gen:*

The key generation algorithm takes as input the public parameters, an identity ID and the Master Secret Key MSK, and outputs a secret key S for the identity ID.

#### 3) *Encryption:*

Consider that there are k messages {M1, M2, ・・・, Mk}. To encrypt the message Mi, the encryption algorithm takes as input the public parameters, the identity ID and the message Mi, and outputs the cipher text= 1, 2, ・・・, k. It sends the cipher texts to the proxy servers.

#### 4) *Query algorithm:*

It takes as input the receiver's identity ID_, the receiver's secrete key and the cipher text, and outputs an authentication information AI. It sends (ID_, AI, CT) to the proxy server. The proxy server redirects (ID_, AI, Ci, 2) to the owner with identity ID.

#### 5) *Permission algorithm:*

This Algorithm checks the authentication information AI. If the receiver is legal, this algorithm takes as inputs the public parameters, the receiver's identity ID_ and the owner's secret key, and outputs access permission (re-encryption key).

#### 6) *Re-encryption:*

The re-encryption algorithm takes as input the public parameters, the receiver's identity ID_, the access permission and the cipher text, and output cipher text.

#### 7) *Decryption:*

There are two algorithms. One is for the owner and the other is for the receiver.

##### a) *Decryption1:*

The owner decryption algorithm takes as input the public parameters, the owner's secret key and the cipher text, and outputs the message Mi.

##### b) *Decryption2:*

The receiver decryption algorithm takes as input the public parameters, the receiver's secret key and the re-encrypted cipher text.

#### 8) *Proxy Re-encryption Algorithm:*

This algorithm is going to take the input as the public parameters, the receivers ID, and Cipher text that may be the owners (or) receivers and gives the output as Cipher text to the receivers with ID's.

#### 9) *DES:*

This algorithm takes a fixed-length string of plain text bits and transforms it through a series of complicated operations into another cipher text bit string of the same length. In the case of DES, the block size is 64 bits. DES also uses a key to customize the transformation, so that decryption can supposedly only be performed by those who know the particular key used to encrypt.

### C. *Methodologies:*

#### 1) *Data Owner Module:*

The user who uploads the file in cloud server is called data owner. The owner has to be registered in private key generator for uploading files. When we are uploading the data, every node can act as data owner and data consumer. Here the data up loader is called as data owner and the data user is called as data consumer. The data owner first takes parameter, identity, and message and encrypt the message and sent to proxy server. Then the proxy server validates the cipher text and store in database.

*2) Private Key Generator:*

This module is used to generate unique secret keys for each registered user. This module is also used for verifying login details of users. Here, first the new data owner registers and then gets a valid login credentials.

*3) Registration/login Module:*

In this module, user registers with private key generator. The PKG returns unique Master secret key to user which will be used to encrypt file. When a user has to upload or download file, he will be verified by PKG using his login details which consists of user ID and password.

*4) Encryption:*

Here file will be encrypted by owner using secret key which is generated using PKG, which is then passed to DES algorithm. This technique is also used by proxy server to re-encrypt file using re-encryption key of owner.

*5) Proxy Server:*

This module is used to save files uploaded by users. Proxy servers store the encrypted data and transfer the cipher text for the owner to the cipher text for the receiver when they obtain access permission (re-encryption key) from the owner. In these systems, proxy servers are assumed to be trusted. They authenticate receivers and validate access permissions. When user wants to download file, the proxy server gets authentication information of user and sends it to file owner. If information is valid, owner gives access permission (re-encryption key) according to the received information.

*6) File Retrieval Module:*

The proxy server transfers the intended cipher text to the receiver using the received access permission. Decryption is done on the receiver side. First receiver decrypts the cipher text using owner's re-encryption key; then again original file is decrypted using secret key.

*7) Mailing:*

In this part a receiver has to first send authentication information to the owner through mailing before downloading file. If authentication is not done, file cannot be downloaded. When access permission (re-generation key) is generated by owner, it is sent to receiver's mail ID. Receiver can login to his mail id and access re-generation key.

*8) Receiver Module:*

The user who downloads file from proxy server is called as the receiver. The receiver has to send authentication information to proxy server which in turn sent to data owner for getting data and access key, if authentication information is invalid, file cannot be downloaded. The receiver authenticates himself to the owner and decrypts the re-encrypted Cipher text to obtain the data. The use case diagram, data flow diagram and activity diagram are shown in figures form respectively.

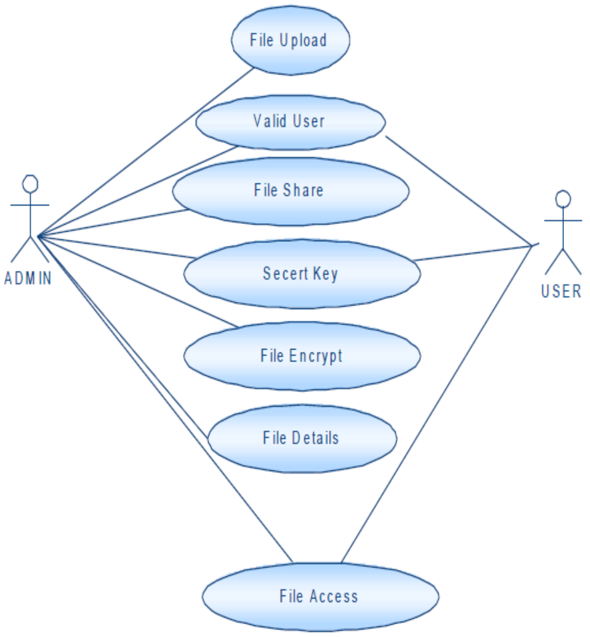

Fig. 4: Use Case Diagram

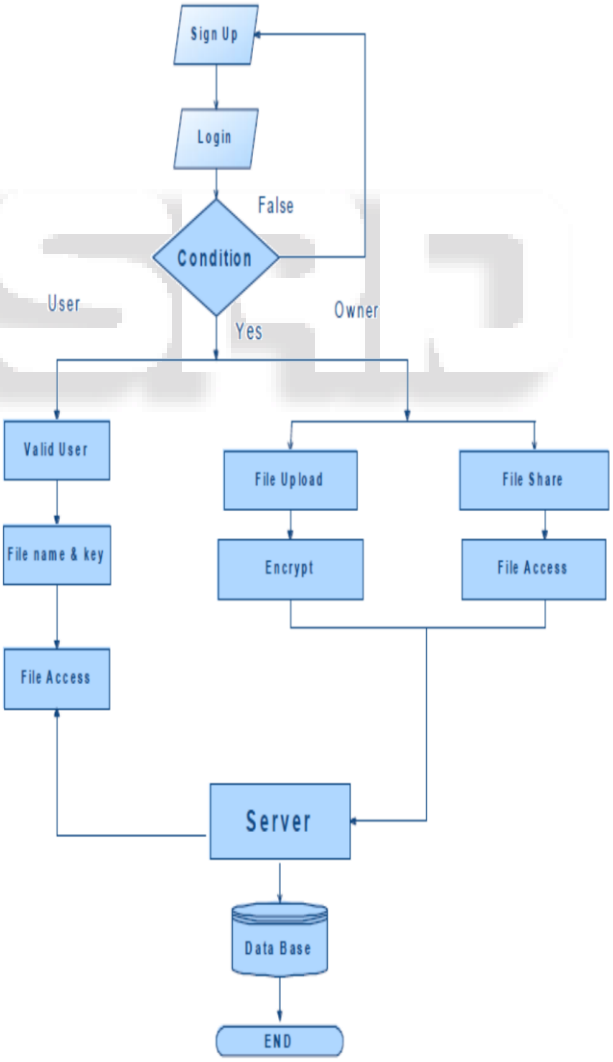

Fig. 3: Data Flow Diagram

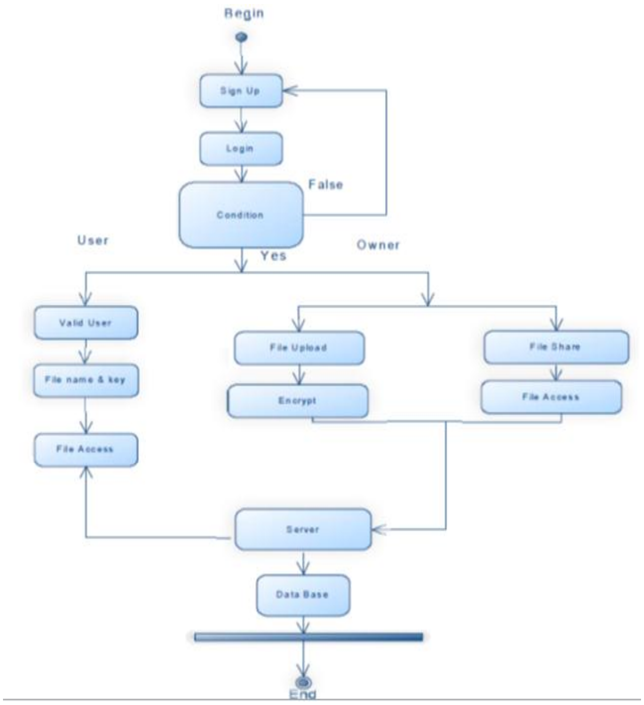

Fig. 5: Activity Diagram

## V. CONCLUSIONS

Distributed data storage schemes provide the users which benefits to outsource their files to untrusted cloud servers. Here, in identity-based secure distributed data storage (IBSDDS) schemes the users are identified by their identities and can communicate without the need of verifying the public key certificates. In this paper, we proposed IBSDDS schemes where, for one request, the receiver can access only one file, instead of all the files of the owner. Furthermore, the access permission can be made by the owner itself, instead of the trusted third party. Noticeably, our scheme is secure against the collusion attacks. Our scheme secure the data against collusion attacks over the previous schemes and will get the notification of user request on their respective mail ID and it provides secure model of cloud storage with safe data forwarding**.**

## REFERENCES

[1] Ivan and Y. Dodis, “Proxy cryptography revisited,” in Proc.Network and Distributed System Security Symposium - NDSS’03, (San Diego, California, USA), pp. 1–20, The Internet Society, Feb. 2003

[2] Armbrust M, Fox A, Grith R, Joseph AD, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I, Zaharia M. A view of cloud computing. Communications of the ACM 2010

[3] Bouganim L, Pucheral P. Chip-secured data access: Confidential data on untrusted servers. In: Proceedings: International Conference on Very Large Data Bases: Morgan Kaufmann 2002: 131-142.

[4] Fernando N, Loke SW, Rahayu W. Mobile cloud computing: A survey. Future Generation Computer Systems 2013; 29(1): 84-106.

[5] Jinguang Han, Willy Susilo, and Yi Mu: Identity-Based Secure Distributed Data Storage Schemes, IEEE Transactions on Computers Vol: Pp No: 99 Year 2013.

[6] Han, J., Susilo, W. & Mu, Y. (2013). Identity-based data storage in cloud computing. Future Generation Computer Systems: international journal of grid computing: theory, methods and applications, 673-681.

[7] L. Wang, L. Wang, M. Mambo, and E. Okamoto, “New identity based proxy re-encryption schemes to prevent collusion attacks,” in Proc. Pairing-Based Cryptography - Pairing’10, vol. 6487r, Dec. 2010.

[8] M. Green and G. Ateniese, “Identity-based proxy re-encryption,” in Proc. Applied Cryptography and Network Security - ACNS’07 , vol. 4521, pp. 288–306, Springer, Jun. 2007.

[9] Qin Liu, Guojun Wang and Jie Wu, “Efficient Sharing of Secure Cloud Storage Services,” 10th IEEE International Conference on Computer and Information Technology, 2010

[10] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, “Improved proxy re-encryption schemes with applications to secure distributed storage,” ACM Transactions on Information and System Security, vol. 9, no. 1, pp. 1–30, 2006.

[11] S. D. C. di Vimercati, S. Foresti, S. Paraboschi, G. Pelosi, and P. Samarati, “Effficient and private access to outsourced data,” in Proc. International Conference on Distributed Computing Systems - ICDCS’11, (Minneapolis, Minnesota, USA), pp. 710–719, IEEE, Jun. 2011.

[12] H.Y. Lin and W. G. Tzeng, “A secure erasure code-based cloud storage system with secure data forwarding,” IEEE Transactions on Parallel and Distributed Systems, Digital Object Indentifier 10.1109/TPDS.2011.252 2012.

[13] H. Shacham and B. Waters, “Compact proofs of retrievability,” in Proc. Advances in Cryptology - ASIACRYPT’08 (J. Pieprzyk, ed.), vol. 5350 of Lecture Notes in Computer Science, (Melbourne, Australia), pp. 90–107, Springer, Dec. 2008.

[14] A. Juels and B. S. K. Jr., “PORs: Proofs of retrievability for large files,” in Proceedings: ACM Conference on Computer and Communications Security - CCS’07 (P. Ning, S. D. C. di Vimercati, and P. F. Syverson, eds.), (Alexandria, Virginia, USA), pp. 584–597, ACM, Oct. 2007.

[15] M. Satyanarayanan,“Scalable, secure, and highly available distributed file access,” IEEE Computer, vol. 23, no. 5, pp. 9–21, 1990.

[16] S. Forrest, S. A. Hofmeyr, and A. Somayaji, “Sense of self for unix processes” in Proc. IEEE Symposium on Security and Privacy - S&P’96 (Oakland, CA, USA), pp. 120–128, IEEE, May 1996.

[17] A. G. Pennington, J. L. Griffin, J. S. Bucy, J. D. Strunk, and G. R. Ganger, “Storage-based intrusion detection,” ACM Transactions on Information and System Security, vol. 13, no. 4, pp. 30:1–30:27, 2010.