Security Framework for Distributed Data Processing

Tatiana Galibus[1], Julio C.S. dos Anjos[2], Edison Pignaton de Freitas[2], Cláudio F. Resin Geyer[2], Gilles Fedak[3], Rafael Tim´oteo de Sousa Jr.[4], João Paulo C. L. Costa[4], Rubem Pereira[5], Paul Fergus[5], Anton Zaleski[6], Herman Vissia[6], Volker Markl[7]

[1]Belarusian State University, 220030 Minsk, 4, Nezavisimosti – Minsk – Belarus
tan2tan@gmail.com
[2]Federal University of Rio Grande do Sul, Institute of Informatics - PPGC - Brazil
{ jcsanjos, edison.pignaton, geyer } @inf.ufrgs.br
[3]INRIA, Avalon - ENS Lyon, France gilles.fedak@inria.fr
[4]University of Brasilia, UnB - FT – ENE – CP: 4386 - 70910-900, Bras´ılia - DF - Brazil
desousa@unb.br, joaopaulo.dacosta@ene.unb.br
[5]Liverpool John Moores University, LJMU City Campus, Liverpool, England
{ r.pereira,p.fergus } @ljmu.ac.uk
[6]Byelex Multimedia Products BV, Argon 1, 4751 XC Oud Gastel, The Netherlands
a.zalesky@by.byelex.com, h.vissia@byelex.com
[7]Technische Universitat Berlin - DIMA - School of EECS - Einsteinufer 17 - 10587 - Berlin – Germany volker.markl@TU-Berlin.de

Abstract: *The growth of importance of distributed data processing services highlights the importance of building security solutions that address the needs of these systems. In this context, this work proposes an innovative approach for implementing a flexible security framework for data processing services in distributed environment. The key features of the proposed framework are the support of the large-scale, medium-sized and small and medium-sized corporate data analytic services. The cryptographic core of the proposed framework is based on the digital signature and a hybrid encryption system using a modified attribute-based encryption. The paper outlines the conceptual architecture of the framework and presents performance testing results.*

Keywords**:** Security, Big Data Security, SMART Security Platform,

1. INTRODUCTION

The distributed data processing owns specific characteristics which distinguish it from other applications, such as, volume, variety, velocity, value and veracity [1]. It demands specific solutions that are able to address these aspects. A large volume of data (volume) from different sources that ingress in the cloud with different formats (variety) are processed in real-time (velocity) with high accuracy (veracity) to find competitive information (value) for a given final user application. Also, the data needs to be homogenized and coherently handled by the cloud or other infrastructures, like hybrid infrastructures [2]. This infrastructure needs to be secured to guarantee the data privacy and avoid misuse of the services it offers. An efficient distributed data security framework should include the following components: Authentication services; Access control infrastructure for the supported access policies; Encryption protocols and other standard cryptography services; and Fast on-the-fly auditing services. All currently used industrial approaches [3] are designed for the Hadoop infrastructure and utilize Kerberos secure resource sharing protocol. Besides, the theoretical base of the big data security is rather poor: there are works related to access control policy [4], hybrid environment machine-learning systems [5] among other, but none of the works found in the literature provide a comprehensive security solution. This fact implies that there are no general approaches in this field.

Therefore, the distributed data processing security theory lacks the formal conceptual models, which leads to the absence of the verifiable solutions for Big Data security [6] and [7]. Trying to cover this gap, this paper outlines the formal conceptual model of the security infrastructure and presents a practical framework to address the above security demands. The methodology supports several cryptographic mechanisms designed for the different levels of controlling the access to sensitive data. The proposal includes a framework called Small & Medium sized Enterprise Data Analytic in Real Time (SMART) [8].

SMART is a modular framework for Big Data analysis that uses clouds or hybrid infrastructures to provide support for Small & Medium-sized services operation. The core security framework is implemented in the protected cloud infrastructure [9]. The rest of this paper is structured as follows. Section 2 examines relevant related works. In Section 3, there is a description overview of the main characteristics of the proposed cloud infrastructure. Section 4 describes the proposed authentication and access control mechanisms for this cloud infrastructure. Section 5 presents performance and security testing results including the basic adversary model. Section 6 concludes the paper and presents suggestions for future work

2. RELATED WORK

The most important problems of securing the distributed data processing are mentioned in the Cloud Security Alliance report [10]. Among them there are: mapper security, middleware protection, end-point security, real-time monitoring, user privacy protection and access control. The most challenging task is the necessity to modify the traditional and conventional mechanisms before actually applying them in the distributed services. The critical components, such as access control, remain elusive [11], [12].

Hadoop-oriented security approaches based on the Kerberos resource sharing protocol [13] turn out to be ineffective in relation to support and implementation [12], [11]. Apart from that, it is difficult, and rather time-consuming, to set up and configure the Kerberos infrastructure [11]. In the cloud computing system, it is preferable to use either public-key infrastructure or attribute-based encryption for protecting the user privacy [14] while the symmetric encryption should be used to support the fast processing of the bulk data [15].

It can be concluded that there is a demand for developing security models and policies, not related to Hadoop security infrastructure based on Kerberos [11], [16], [17]. Such models allow to analyze the consistency of the proposed security architecture, and figure out the basic adversary model [16]. Other can be applied for distributed data processing infrastructures, such as Flink, possessing more flexibility in providing various data processing functions [2].

According to the CSA report [10], the most promising cryptographic tool for security framework implementation is the Attribute-based Encryption (ABE) [3], [19], [18]. The most important challenge related to ABE implementation is the necessity to apply the modifications in order to support the key revocation [20] and simplify the key generation [14].

In the light of these problems, this study provides a security model corresponding to the attribute-based access model. The corresponding implementation design is provided in order to support the model. This work aims not only to construct the model [11], [16], [17], but that provides a practical ABE-based frameork design and implementation.

3. CLOUD INFRASTRUCTURE

3.1 Architecture Overview

The complex cloud services need to be configured automatically at several abstraction levels. Cloud infrastructures comprising heterogeneous hardware environments may need the specification of configuration parameters at several levels such as the operational system, service containers and network capabilities [21]. Figure 1 illustrates the solution proposed to model a hybrid system which depicts a Global Dispatcher and Global Aggregator to be used on the infrastructure. As the systems are independent different data sizes can be manipulated at same time. Previous work has been performed on MapReduce for hybrid environments [2], [22].

The Global Dispatcher located outside the cloud has middleware functions for handling task assignment, and management of user-provided data. It is a centralized data storage system that manages policies for split data and distribution in accordance with the needs of each system.

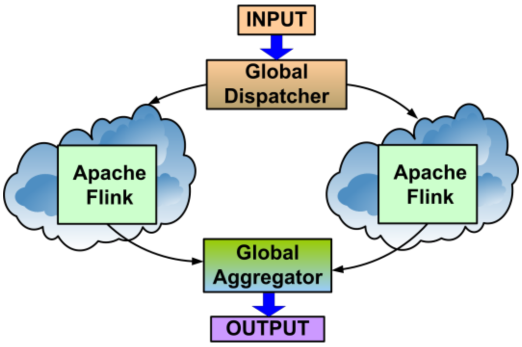


Fig. 1 — SMART architecture

The principle is similar to a publish/subscribe service (PS-S) in which the system obtains data and publishes computing results [2]. PS-S is a mechanism producer/consumer that works like a Queue Message. The user defines tasks and local data for put in a queue over Apache Kafka in similar manner to [23] and after the scheduler from global dispatcher distributes this tasks in the queue for machines in the Cloud. The Global Aggregator data output from both systems and merges them in order to obtain the final dataset.

4. FORMAL SECURITY MODEL

The proposed approach to secure the distributed data processing on the cloud is an extension of a cloud-based access control and privacy protection infrastructure which is currently implemented in protected enterprise cloud storage- Storgrid [9]. The core of the protected environment in the cloud is the hybrid attribute-based encryption [14] with additional parameters which allow to support the protection of heterogeneous devices and attribute access policy avoiding the immense use/optimizing the use of computational resources. Therefore, it is suitable for the SMART cloudbased architecture and corresponding Big Data processing services. A simpler method of securing the user-generated data is used for the services the require the rapid data processing. In other words, the Storgrid security framework is extended with the following infrastructure, shows in Figure 2:

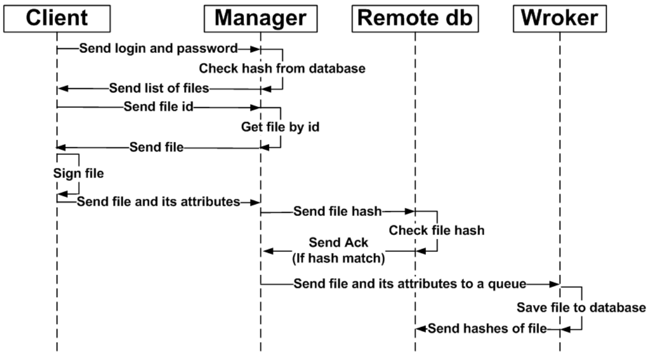


Fig. 2 — SMART secure method

The protection of the services in such system is is based on the two basic mechanisms: 1) digital signature i.e.no one should be able to modify the data entered by a specific user other than the user himself and 2) the selective cryptography i.e. only the specific pieces of information collected are encrypted in order to improve the quality of sort and search service operations.

In order to protect the privacy and provide security for the enterprise services the initial Storgrid hybrid attribute-based AC is used. In this case, the performance is compromised for the sake of better security.

The proposed approach is based on several cryptographic mechanisms. The files/bulk data/multimedia are encrypted with 128/256-bit AES, while the permanent file keys are encrypted with the attribute-based encryption. The set of expiring ABE keys corresponding to the set of files accessible by user in encrypted with a single expiring AES key (KEY SET KEY). This key is split by server into four stage (two are stored on the device and two belong to the user) by the secret sharing scheme (SSS). The encryption workflow is outlined in the following Figure 3.

With each user session, the permanent FILE KEY (unique AES key) is re-encrypted. The set of FILE KEYs is protected with the corresponding ABE keys. The unique ABE model supports the attribute policy based on user groups and on file shares. The model supports the simple selective ABE scheme [28], [29]. The selective scheme for attribute-based encryption is as follows: if at least one attribute in the set $\{t_i\}_U$ is equal to the attribute in the set $\{t_i\}_M$, the corresponding user U can decrypt the text M . In other words, as soon as the user and share have one attribute in common - the user can get access to the share.

The components of the ABE encryption are:

- Master-key (MK) which is kept safely on server and accessible only for the domain administrator and is specified by MK = $(t_1, t_2, ..., t_n , y)$, where the values t_i are randomly selected from the group Z_p . They are the private keys corresponding to the group attributes. Note, that this is different from the usual PK encryption: the private keys are controlled by the admin and not by the users.
- Public key (PK) depends on the master key values and is kept in the clear allowing users to access the information: PK = $(g^{t\,1}, g^{t\,2}, ..., g^{t\,n}, .., e(g, g)^y)$, Here e(g, g) is the bilinear pairing function corresponding to an elliptic curve.

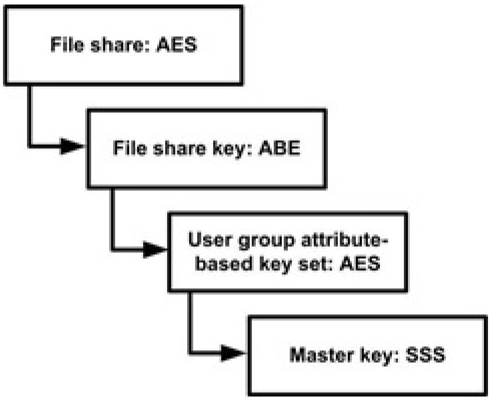


Fig. 3 — Encryption workflow

- Secret user KEY SET depends on his attribute set. Here each D_i (GROUP KEY) serves for decryption of the data of a single group of users, for example, related to some project: $\{t_i\}_U \rightarrow D = \{D_i = g^{yw/ti}\}$.
- Encrypted text M , in our context, M=FILEKEY, or the permanent AES symmetric key, which allows to avoid the file re-encryption.

Encryption procedure is multiplication. The set of the public keys E i (PUBLIC SHARE KEY) corresponding to the set of groups able to access the text is kept along with the encrypted text E :

$$E = M\, e(g, g)^y s, \{E_i = g^{t\,i\,s/w}\}, i \in \{t_i\}_M$$

Decryption is division: $M = E/Y^s$

In order to perform this operation the user needs the pair of private key D_i and public key E_i corresponding to the attribute t_i : $Y^s = e(g, g)^{ys} = e(E_i , D_i) = e(g^{yw/t\,i}, g^{t\,i\,s/w}) = e(g, g)^{ys}$

The result of decryption is the FILE KEY - the symmetric AES key that permits to decrypt the contents of protected file.

The proposed security mechanisms are implemented based on the following infrastructure in Figure 4.

Encryption server manages all the certified authentication (CA) and encryption operations and grants the user access to the data storage. This server can store the encryption keys and/or connect to a separate Key Storage server. This server generates the user keys, connects to the client UI, runs the key renewal routines, stores the user public keys and attributes besides the auditing data.

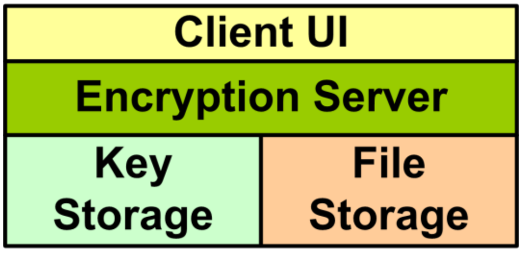


Fig.4 — Security Infrastructure Components

File storage is secure in the sense that some of the files specified by the domain administrator are stored, encryptedand have restricted access. It is recommended to encrypt this external part of file storage completely.

Client UI can connect to the Encryption server and ask for the permission to access the file storage in order to view/edit/upload specific files or folders. Client UI stores the user keys for the ABE encryption and the unique symmetric session keys which serve for restricting the access to the downloaded files. The symmetric keys are encrypted with the ABE keys. The client supports different platforms and operating systems.

The hybrid encryption method allows controlling the privacy of the users without compromising the overall encryption time. The basic ABE [18], [19] approach was modified in order to set up the validation period from user key and support the attributes corresponding to both the file shares and user groups.

In the following section, the proposed security infrastructure is integrated into the Big Data environment.

4.1 Security Framework for SMART Cloud-Based Processing Service

The functions of the encryption server, *i.e.*, the protection services that work once the data is uploaded (authentica tion, CA, encryption) need to be separated.

Figure 5. shows six modules Global Collector, Global Dispatcher, Storage, Core Engine, Global Aggregator and Central Monitoring.

The *Core Engine* must support hybrid systems, *i.e.*, enable streaming and batch computations at the same time. Therefore, the Flink framework is an important system to consider. The MR-BitDew [24] is another framework to improve computational performance, withthe Volunteer Computing use in a hybrid infrastructures. A Client User API provides an easy method to the users submit their applications and indicate the data sources. The Client UI is a security interface that enables a single user identification through an encrypted key. A key is employed in the Encryption-Decryption Engine.

The *Global Collector* layer maintains management and coordination of sensing modules. It is responsible for getting data from several sources and maintaining the data integrity mechanisms. The data is collected and serialized under a standard TCP/IP, which composes the communication stack to the *Global Dispatcher*.

In the *Global Dispatcher*, the data is decoupled of the lower layers in the message queue mechanism. The data is put in a FIFO queue to be distributed to severs accordance with their resources availability in both *Cloud/Multi-Cloud* and *Grid/Multi-Grid* environments. Optimization layer analyzes the input data volume and decides through *Decision Engine* about scheduling tasks and data through distinct environments. A simulation process implements an execution time prevision that will be used by *Decision Engine* to better the accuracy of scheduling mechanism. The user deploying module enables unsafe and encrypted data localization under external server. The user must provide a key storage localization and the data path across the network before definitely attaching the data. The storage-and-forward and pass-through protocols are implemented inside this layer according to data source.

The intermediate results, processed on *Core Engine*, are serialized to *Global Aggregator* that have the data consolidation mission. The *Data Integration* module does support the data integrity and data integration. The last phase of data processing is designed to generate an iterative execution and provide the result consolidation. A *Communication API* is necessary to integration the workers into a virtual network to data computation. The *Aggregator API* is a module that orchestrates the results aggregation and maintains the safety data mechanism for end-users. The *End-User Interface* shows the information with a friendly visual characteristic through a central monitoring.

This integration model makes it possible to implement the required level of security and user privacy in all types of governmental and corporation organizations and services. The flexible implementation design allows protecting the user privacy and control the access to the sensitive data in the heterogeneous environment via the *Client UI* developed for various OS and various platforms. This architecture can be easily extended in order to use more sophisticated methods of ensuring the data and key protection, *i.e.*, secret sharing or complex attribute policies.

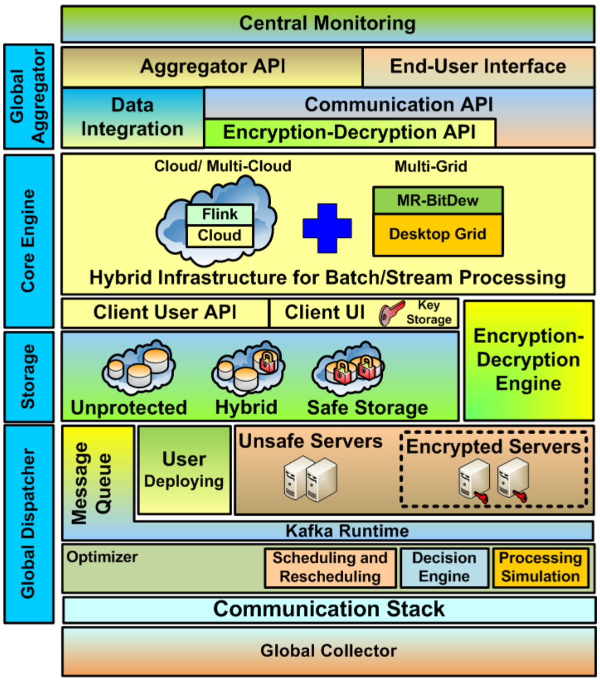


Fig.5 —Components of the security infrastructure and their interactions

5. IMPLEMENTION AND PRELIMINARY RESULTS

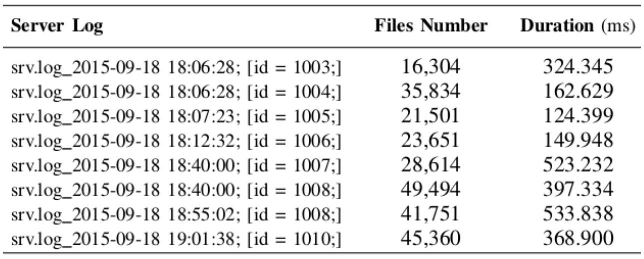
The proposed security infrastructure is currently being implemented as part of the Storgrid project. Storgrid is a secure cloud storage management system. This section presents some results and analysis of the security framework similar to the presented model. The administrator (or domain manager) controls the server work, and has the access server through the secret password. Their responsibility is to decide the level of security of the protected/public files and define an authorization policy on a web interface.

The basic encryption method on the server side is 128 bit AES. In order to enhance the security, the file storage secret key has expiry period. The procedure of key regeneration and re-encryption occurs daily and does not affect the performance because the procedure runs when the system is idle or has low activity. Auditing and log analysis performed on server side do not allow to perform an attack on a key.

The preliminary results of the re-encryption testing taken from the server real-time log are shown in Table 1. It demonstrates that the average encryption time even for the big bulk of data, compared to the size of the data collected by the SMART services, is rather low. The encryption time depends on the server load at the time when encryption has been performed. The results demonstrate a low overhead to the protected file re-encryption in local file system.

Therefore, this re-encryption mechanism can be successfully integrated into the existing SMART infrastructure. Other reason to use re-encryption is that presented model does not use the concept of complete data encryption. This provides a selective encryption of the sensitive user data in order to avoid overhead. This concept allows does not overload internal network and controls the security of the files. The control of this process is performed by the domain manager access control utility.

Table 1. — Re-encryption time results



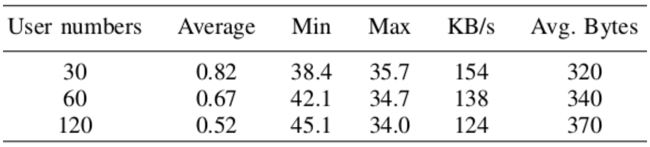
Server Log	Files Number	Duration (ms)
srv.log_2015-09-18 18:06:28; [id = 1003;]	16,304	324.345
srv.log_2015-09-18 18:06:28; [id = 1004;]	35,834	162.629
srv.log_2015-09-18 18:07:23; [id = 1005;]	21,501	124.399
srv.log_2015-09-18 18:12:32; [id = 1006;]	23,651	149.948
srv.log_2015-09-18 18:40:00; [id = 1007;]	28,614	523.232
srv.log_2015-09-18 18:40:00; [id = 1008;]	49,494	397.334
srv.log_2015-09-18 18:55:02; [id = 1008;]	41,751	533.838
srv.log_2015-09-18 19:01:38; [id = 1010;]	45,360	368.900

The client controls security from the user side and in the cloud. It is implemented on different platforms. The proposal supports the concept of the light-weighted client,

i.e., it does not do much encryption or other critical/heavy operations. All encryption is done on the server side and client just supports the key usage. The unique feature is that every time the encrypted files are not downloaded again if the keys remain the same. The user can use his downloaded file after authentication with the password and email and it is kept securely (encrypted with unique AES key) in his storage device. The access to the AES key is controlled by the server, ABE encryption and key expiry period. The operation of regenerating the public keys and resending them to users are not time-consuming due to the implementation design. The results of testing the user key generation procedure for the 1000 user logins and 1000 key samples out of 20 parallel threads are show in Table 2.

These results show that the proposed security framework is designed to reduce the network load without compromising the level of user security. These properties perfectly match the security needs of the SMART architecture.

Table 2. — User Key Generation Procedure Results



User numbers	Average	Min	Max	KB/s	Avg. Bytes
30	0.82	38.4	35.7	154	320
60	0.67	42.1	34.7	138	340
120	0.52	45.1	34.0	124	370

6. CONCLUSION

This paper proposes a security framework for distributed data processing services of a different scale. Particularly, this proposal aims cloud-based systems that handle massive amounts of data in real time analysis coming from very diverse data sources, which is the motivation for the SMART architecture. The proposed model provides authentication and access control mechanisms appropriate to deal with the high demands of the applications intended to be support by SMART. The framework testing provides evidence that it imposes low overhead without compromising the security level being suitable for SMART application purposes.

REFERENCES

[1] M. Stonebraker, S. Madden, and P. Dubey, "Intel "Big Data" Science and Technology Center Vision and Execution Plan," *SIGMOD Rec.*, vol. 42, no. 1, pp. 44–49, May 2013. [Online]. Available: http://doi.acm.org/10.1145 /2481528.2481537

[2] J. C. S. Anjos, G. Fedak, and C. F. R. Geyer, "BIGhybrid: a simulator for MapReduce applications in hybrid distributed infrastructures validated with the Grid5000 experimental platform," *Concurrency and Computation: Practice and Experience*, vol. 1, no. Special Issue Paper, pp. 1–24, September 2015, cpe.3665. [Online]. Available: http://dx.doi.org/10.1002/cpe.3665

[3] Z. Qiao, S. Liang, S. Davis, and H. Jiang, "Survey of attribute based encryption," in *Software Engineering, Artificial Intelligence, Networking and Parallel/ Distributed Computing (SNPD), 2014 15th IEEE/ACIS International Conference on*, June 2014, pp. 1–6.

[4] V. C. Hu, D. Ferraiolo, R. Kuhn, A. Schnitzer, K. Sandlin, R. Miller, and K. Scarfone, "Computer Security," *Tech. Rep.* 800-162, Jan. 2014, accessed in September 2015. [Online]. Available: http://dx.doi.org/10.6028/NIST.SP.800-162

[5] J. Whitworth and S. Suthaharan, "Security Problems and Challenges in a Machine Learning-based Hybrid Big Data Processing Network Systems," *SIGMETRICS Perform. Eval.* Rev., vol. 41, no. 4, pp. 82–85, Apr. 2014. [Online]. Available: http://doi.acm.org/10.1145/ 2627534.2627560

[6] Q. Liu, C. C.Tan, J. Wu, and G. Wang, "Reliable Re-Encryption in Unreliable Clouds," in *Global Telecommunications Conference (GLOBECOM 2011)*, 2011 IEEE, Dec 2011, pp. 1–5.

[7] D. Chen and H. Zhao, "Data Security and Privacy Protection Issues in Cloud Computing," in *Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on*, vol. 1, March 2012, pp. 647–651.

[8] J. C. S. Anjos, M. D. Assuncao, J. Bez, C. F. R. Geyer, E. P. de Freitas, A. Carissimi, J. P. C. L. Costa, G. Fedak, F. Freitag, V. Markl, P. Fergus, and R. Pereira, "SMART: An Application Framework for Real Time Big Data Analysis on Heterogeneous Cloud Environments," in Computer and Information Technology; *Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*, Oct 2015, pp. 199–206.

[9] B. D. S. BV, "Storgrid EFSS: Secure Enterprise File Sharing Software," Jan. 2016, available from Internet: http://www.storgrid.com/.

[10] S. Rajan, W. V. Ginkel, N. Sundaresan et al., "Expanded Top Ten Big Data Security and Privacy Challenges," Tech. Rep., Apr. 2013. [Online]. Available: https://downloads.cloudsecurityalliance.org/initiatives/bdwg/Expanded Top Ten Big Data Security and Privacy Challenges.pdf

[11] V. C. Hu, T. Grance, D. F. Ferraiolo, and D. R. Kuhn, "An Access Control scheme for Big Data processing," in *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 International Conference on.* IEEE Computer Society, Oct 2014, pp. 1–7.

[12] B. Lublinsky, K. T. Smith, and A. Yakubovich, *Professional Hadoop Solutions,* 1st ed. 10475 Crosspoint Boulevard: John Wiley & Sons, Inc., Sep. 2013

[13] K. Zheng and W. Jiang, "A token authentication solution for hadoop based on kerberos pre-authentication," in *Data Science and Advanced Analytics (DSAA), 2014*

International Conference on, Oct 2014, pp. 354–360.

[14] T. Galibus and H. Vissia, "Cloud storage security," in *Network Security and Communication Engineering*, K. Chan, Ed. CRC Press, Jun. 2015, pp. 123–126.

[15] U. S. of Commerce, "Announcing the Adcanced Encryption Standard (AES) ," (NTIS), 5285 Port Royal Road, Springfield, VA 22161, Tech. Rep., Nov. 2001, accessed in September 2015. [Online]. Available: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[16] V. C. HU, D. R. KUHN, T. XIE, and J. HWANG, "Model Checking for Verification of Mandatory Access Control Models and Properties," *International Journal of Software Engineering and Knowledge Engineering*, vol. 21, no. 01, pp. 103–127, 2011.

[17] W. Zeng, Y. Yang, and B. Luo, "Access control for big data using data content," in *Big Data, 2013 IEEE International Conference on*. IEEE Computer Society, Oct 2013, pp. 45–47.

[18] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute Based Encryption," in *Security and Privacy, 2007. SP '07. IEEE Symposium on*, May 2007, pp. 321–334.

[19] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based Encryption for Fine-grained Access Control of Encrypted Data," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, ser. CCS '06. New York, NY, USA: ACM, 2006, pp. 89–98. [Online]. Available: http://doi.acm.org/10.1145/1180405.1180418

[20] Z. Liu and D. S. Wong, "Practical Attribute-Based Encryption: Traitor Tracing, Revocation, and Large Universe," *Cryptology ePrint Archive, Report* 2014/616, 2014, http://eprint.iacr.org.

[21] D.-H. Le, H.-L. Truong, G. Copil, S. Nastic, and S. Dustdar, "SALSA: A Framework for Dynamic Configuration of Cloud Services," in *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on*, Dec 2014, pp. 146–153.

[22] S. Delamare, G. Fedak, D. Kondo, and O. Lodygensky, "SpeQuloS: a QoS service for BoT applications using best effort distributed computing infrastructures," in *Proceedings of the 21th international symposium on High-Performance Parallel and Distributed Computing*, ser. HPDC '12. New York, NY, USA: ACM, 2012, pp. 173–186. [Online]. Available:http://doi.acm.org/ 10.1145/ 2287076.2287106

[23] T. Zhang, "Reliable Event Messaging in Big Data Enterprises: Looking for the Balance Between Producers and Consumers," in *Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems*, ser. DEBS '15. New York, NY, USA: ACM, 2015, pp. 226–233. [Online]. Available: http: //doi.acm.org/10.1145/2675743.2771878

[24] L. Lu, H. Jin, X. Shi, and G. Fedak, "Assessing MapReduce for Internet Computing: A Comparison of Hadoop and BitDew-MapReduce," in *Proceedings of the 2012 ACM/IEEE 13th Int. Conference on Grid Computing*, ser. GRID '12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 76–84.