Big Data Security: Requirements, Challenges and Preservation of Private Data inside Mobile Operators

Cem Dincer[1] and Engin Zeydan[2]
[1]Security Planning and Application Department
Türk Telekom, Istanbul, Turkey
E-mail: cem.dincer@tuktelekom.com.tr
[2]Türk Telekom Labs, Istanbul, Turkey
E-mail: engin.zeydan@turktelekom.com.tr

***Abstract*—Today's mobile operators (MOs) are experiencing tremendous amount of data usage of their subscribers. This results in data tsunami arriving from various sources inside MOs' network infrastructures. On the other hand, handling this data increase in an elegant manner will be utmost importance for designing the next generation cellular 5G network infrastructure. In this evolving 5G architecture, there will also be many vertical market players from different domains (e.g. car manufacturers, retailers, banks, transportation providers) as well as third party players (e.g. application developers) that will be interacting with MO's subscribers private data through many innovative big data applications. These big data applications can provide additional value added services to MOs. On the other hand, ensuring the security of these applications utilizing big data will be another dimension that needs to be provisioned carefully. In this paper, we study the various security requirements and challenges of running big data application services by MOs themselves. Based on the system architecture that is studied, we have also run extensive vulnerability tests against one example of big data application deployed inside MO's premises. Our results indicate key security findings and map some of the missing requirements into the considered big data application scenario.**

***Index Terms*—big data security, cellular networks, mobile operators, private data.**

I. Introduction

Big data is a new phenomenon where massive amounts of data (e.g. system, security, application, traffic logs and others etc) is generated by end hosts (computers, mobile devices, etc) and network devices (e.g. firewalls, routers, etc) inside a Mobile Operator (MO)'s infrastructure. This sheer amount of data is huge and high volume of data, high velocity of data transmission and high variety of data contents are some of its features, which is also abbreviated as 3Vs of big data [1] [2]. As an example, in terms of volume, global mobile data traffic is expected to increase to $24,314$ Petabyte per month in 2019 which is an almost ten-fold increase compared to 2014 [3].

Telecommunications operators have access to vast amount of subscriber related information in the form of Call-Detail Records (CDR), demographic information (age, gender, occupation, etc), usage behaviour of customers (data bandwidth usage, communication services usage, etc) and geo-location data. Indeed, inside every MOs, there are multiple objectives and functionality of multiple departments that need to be aligned in order to leverage the value of this data. For example, there are different vertical use cases (e.g. in smart city [4] or e-health [5] domains) where marketing departments are seeking ways to monetize this information by selling it to third parties through usage of some big data applications or as part of value added services such as location-specific coupons. On the other hand, regulation, security and legal departments of MOs are willing to protect the core business against risks of failing reputation and abuse.

Traditional relational database management systems (RDMS) systems can provide extensive security features such as data encryption, management of user configurations, authentication and access controls, etc. On the other hand, big data solutions have mostly focused their efforts on building efficient data processing pipelines rather than dealing with security aspects [6]. Therefore, compared to the traditional RDMS systems, big data or Hadoop-based systems are still undergoing major upgrades/updates. Some of the security features of Hadoop consist of authentication (e.g. for Hypertext Transfer Protocol (HTTP) web clients), service level authorization (e.g. for task authorizations using job tokens, allowing a configured list of users/groups to submit jobs, etc), access control lists (file permission checks for Hadoop Distributed File System (HDFS)) and data confidentiality (e.g. through network encryption) [7]. Data analytics can also be applied for revealing the intrusions, denial of service attacks and fraudulent events in security applications [8].

Even though big data enables the opportunity for collecting more data, it also brings additional security challenges in terms of privacy and security [9]. Security of big data applications is important in telecommunication industry because of various factors such as: (i) Existence of laws and regulations for ensuring data privacy, (ii) protecting proprietary data of MO's subscribers, (iii) Internal teams in a MO may need different portions of data depending on the MO policies, (iv) Multiple clusters set-up inside the MO for various purposes may or may not contain protected data depending on the demands of various internal departments. Hence, some of the key

questions that need to be addressed for data security of big data applications deployed inside MO's premises are:

- Due to sheer amount of data to review, how do the businesses running big data queries over the applications should access the data? What functionality need to be accessible when accessing this data while maintaining data confidentiality?
- Where does the big data cluster need to be placed inside MO's architecture in order to provide secure service to enterprise users and third party clients?
- What challenges exist in the case when MOs want to build their own cluster and open a service utilizing their own big data through a big data application?

A. Related Work

In the literature, there are many works related to big data security, but still not many articles are discussing about security issues of big data applications deployed inside the premises of MOs. In [8], the authors have outlined some of the open issues and challenges with big data analytics for information security. In [10], the authors have outlined all the security solutions to secure the Hadoop ecosystem. Top ten big data-specific security and privacy challenges are highlighted in [11] by the Cloud Security Alliance (CSA) Big Data Working Group for ensuring computing infrastructure for big data processing is more secure. The authors in [9] have identified big data privacy requirements and discussed the adequacy of existing privacy-preserving techniques for privacy challenges in the era of big data. The authors in [12] have provided a state-of-art overview of recent research issues and achievements in the field of privacy and security of big data. The whitepaper in [13] has outlined the security gaps involved in open source Hadoop distributions and evaluated the paths being taken by different Hadoop distribution and data security vendors.

In addition to above works, there are various different open source, common off the shelf (COTS) big data security or "ecosystem" tools as well that can interact with Hadoop and be used within a Hadoop big data solution [14] [15] [16] [17] [18]. Some of them require minor configuration changes for obtaining a secure cluster environment. However, each tool provides different configuration details. Due to security issues of Hadoop-based big data solutions, Kerberos was proposed as the authentication mechanism for Hadoop-based services on all remote procedure calls (RPCs) which is a standard network security protocol [19]. Apaches Sentry is an open source big data security software [14]. Apache Sentry is used to enforce fine-rained role-base authorization as well as multi-tenant administration to data and meta-data stored in HDFS. Project Rhino is an open source Hadoop project thats trying to enhance data protection of the Hadoop ecosystem [15]. Another security solution offering additional security is again an Apache project called Apache Accumulo [16]. Different secure versions of Hadoop distributions are also provided by Cloudera Sentry (now Apache Sentry [14]), DataGuise for Hadoop [18] and DataStax Enterprise products [17]. However, note that depending on the Operating System (OS), the Hadoop distribution version and the related big data applications, Hadoop security still needs to be customized for these tools especially for specific usage inside MOs.

B. Our Contributions

There are not so many different studies focusing on security issues of big data analytics and their applications especially in the context of MOs. In this paper, our first main focus is investigating the security requirements and challenges of running big data applications inside the premises of MOs. The second motivation is to provide a discussion on the potential shortcomings observed in an example big data application in production stage and share the results of vulnerability tests run against this example big data applications inside MOs. The contributions of the paper can be summarized as below:

- Identifying big data security requirements of applications utilizing big data inside MOs.
- Identifying major issues and challenges of securing big data applications in MOs and describing potential solutions.
- Using an example big data application scenario, running extensive vulnerability tests against big data application deployed in MO's premises and investigating its security performances.

The remaining part of the paper is organized as follows: Section II discusses about the system architecture of the example big data application. Section III discusses the big data security requirements and key security challenges of big data applications inside MOs. Section IV shows the results of vulnerability tests performed against the studied big data application. Finally, Section V gives the conclusions and future work.

II. System Architecture and Key Elements of Big Data Application

In this section, we will be presenting an example big data application where the system architecture as well as the elements of the big data application are explained in detail. The architecture diagram showing the big data processing topology and one of the data visualization application deployed inside MO is depicted in Fig. 1. This application can be used for providing visualization services to vertical and third party players.

In this architecture, there are three main units: **CDR and Extended Data Record (XDR)** (XDR denotes a variety of detail records including voice, text, data usage, multimedia messaging service, etc.) **processing unit**, **Open Source Big Data Platform unit** which has the *big data visualization application* and **Operator Data Warehouse**. XDR and CDR processing unit collects the necessary call related data (including voice, data and Short Message Service (SMS) for mobile originated and mobile terminated calls) from network infrastructure elements and periodically posts them into a XDR folder. The mobile network infrastructure consists of various elements including base stations, core network gateways and charging elements. The collected data in the XDR folder is

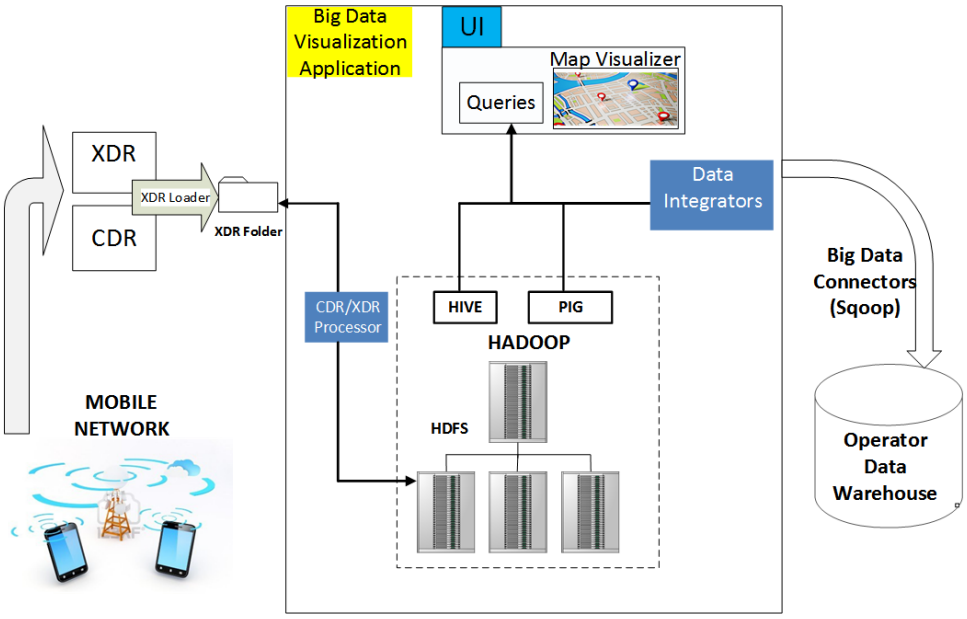

Fig. 1: General architecture diagram of one example of an big data application platform inside a MO.

processed by CDR/XDR processor of big data visualization application for periodic (daily or hourly) imports into HDFS of Hadoop cluster. CDR/XDR processor is an application and performs compression and import of CDR data into the Hive table format defined previously inside big data visualization application. The user interface (UI) interface inside the big data visualization application allows mainly two functionalities: First, it allows querying the Hadoop infrastructure through Hive and Pig languages so that the business analysts or any 3'rd party users may want benefit from the application with appropriate business analysis. An example query that may be of interest would be to correlate users that are in specific locations with their demographic information. Second it allows visualization of the query results on a map, e.g. through Google maps Application Programming Interfaces (APIs). For example, through this map visualizer, the big data application users may visualize the heat map of their query results or they may want to plot the histogram of recent visitors to a specific location during weekdays of a month. The last unit is Operator Data Warehouse where the stored big data in HDFS is imported into the relational database systems through data integrators such as Sqoop. The operator data warehouse is the central database where business analyst run their queries using traditional RDMS. In the next section, we discuss some of the key requirements and challenges of running a big data application similar to the considered system architecture of this section.

III. Big Data Security Requirements and Challenges For Mobile Operators

In this section, we detail some of the key security requirements and challenges of providing big data services via a big data application deployed inside the premises of MOs. As a matter of fact, there exists a trade-off between security and performance for any type of general security solutions. Hence, it is important to know security requirements and challenges of applications beforehand for MOs and make certain that they conform to MOs' security perspective with the appropriate control actions. For example, MOs can utilize existing native Hadoop security solutions, but they may not adapt to solve all the problems that may arise based on the big data application inside MOs.

A. Key security requirements

Some of the key security requirements that are relevant to Hadoop and big data applications deployed inside the premises of MOs can be listed as follows:

1) Architectural Positions and Security of Network Connection Aspects: This requirement is used to ensure that the Hadoop cluster and big data applications' secure location requirements are met in architecture.

- **Cluster Isolation:** The security of a cluster should be ensured by creating its own network subnet.
- **If big data application users want to access to the MO resources over public domain (such as Internet):** Accessing into MO resources from public domain requires three-tier level of secure connection at the following layers i) UI Layer ii) Application Layer iii) Database (DB) Layer. First, the connection requests of outside users originating at UI Layer should terminate in secure gateway at demilitarized zone (DMZ) and second, a new connection should be opened at Application Layer inside MO's Local Area Network (LAN). Finally, the Application Layer should be able to access the DB Layer in LAN. Note that one of the important aspect that needs to be considered is that direct access into the the DB Layer in LAN from UI Layer or DMZ should not be allowed.
- **Integration with existing infrastructure of MOs:** Hadoop security mechanisms need to be coupled with

other network infrastructure e.g. integrating with identity and access management infrastructure of MOs, etc.

- **If big data applications or servers need public connectivity:** In that case, the application should be able to access to Internet through proxy servers or via appropriate firewall permissions re-definitions.
- **If big data applications require integration with a third party system:** The best option for possible integration of big data application is through establishing a lease line connection. If this cannot be achieved, Internet Protocol Security (IPsec) Virtual Private Network (VPN) connection is an alternative solution.

2) Access Control: This requirement is used to ensure that the access and authorization into the big data application is restricted to only authorized users/parties.

- **Authentication:** This is used for confirming the identity of the participant. It can be done via different ways by checking the credentials such as username/password. In a given Hadoop cluster, user authentication is required for all RPCs. Inside MOs, if the authentication of Hadoop users takes place over a user interface via *public domain access*, two-factor authentication methods (such as Username/password plus SMS One-Time Passwords (OTP), Mobile Signature, etc.) should be used over encrypted connections (e.g. Hypertext Transfer Protocol over SSL (HTTPS)). If authentication of Hadoop users takes place over a user interface via *MO's LAN,* integration with central authentication mechanisms such as MO Active Directory should be employed over encrypted connections such as HTTPS. The authentication status should also be cached in order to not explicitly submit credentials at every request.
- **Authorization:** This is used for determining whether an action is allowed to be performed by a Hadoop application user. It is typically done via checking against an access control list. Solutions such as MIT's Kerberos Key Distribution Center (KDC) [20] or Microsoft's Active Directory can provide authorization as well as authentication services.
- **Third Party Access:** In case 3'rd parties access into big data application, access over Internet should be established with VPN account to VPN equipment, and then the connection from VPN device to system should be established with system user. Access to the network/nodes should be limited for untrusted parties.

3) Encryption: This requirement is used to ensure that only authorized users can access a data set in order to view, use or contribute. Three different data protection levels can be used within Hadoop:

- **OS file system-level encryption:** This level of protection should provide a proxy layer between the big data application and the file system that takes into account the trade-offs between performance and impact of encryption. For example, Linux file system level data protection should be capable of encrypting cluster data in and out of HDFS without any modification to application code. Moreover, HDFS encryption keys, certificates and keys need to stored separately from the encrypted data.
- **HDFS level encryption:** The data that rests at the cluster should remain encrypted while it is read from and written to HDFS. In general, HDFS file ownership and permissions are mainly intended for guarding against accidental deletions and overwrites. Therefore, user/group authentication should also be ensured at HDFS folder level where needed. The HDFS client should handle data encryption and decryption.
- **Network level encryption:** The data sent across a network should be encrypted before being sent and decrypted as it is received in order to prevent snooping. For instance, The HDFS data transfer that can occur during intermediate shuffle and sort of map-reduce between the data nodes as well as between data nodes and clients should be encrypted HTTP traffic. Moreover, if a Hadoop Web UI exists for a big data application, access into it should be done via encrypted HTTP traffic. This protection should be ensured using industry-standard protocols such as Secure Socket Layer (SSL)/Transport Layer Security (TLS)

4) Data Confidentiality: This requirement is used to ensure that the confidentiality of customer-related data as well as their location information are kept during external/internal queries and other related information searches.

- **Data confidentiality of customer information:** Data regarding these MO's customer specific information (e.g. Mobile Station International Subscriber Directory Number (MSISDN), International Mobile Station Equipment Identity (IMEI) numbers, CDR and demographics data, etc) should be encrypted using encryption algorithms according to MO's policies in case transfer or storage of these information is performed. The encrypting key and the encrypted information should reside in different places. Moreover, while performing job processing, data decryption should not lag since small amount of lag can have an huge impact across the cluster nodes.
- **Password transmission inside the network:** The passwords used across the network should be used to generate encryption keys.

5) Privacy-Preserving: This requirement is used to ensure that the privacy-preserving techniques are applied to the big data set of MO's subscribers that will be used throughout the big data application or shared with third-parties for further analysis.

- **If big data applications allow location based analytic queries of third parties:** Big Data applications deployed inside Hadoop cluster must be able to process queries from multiple clients in a privacy-preserving manner, i.e. the list of customers' locations should not be shared or inferred from data while allowing other businesses to obtain useful information out of it. For this, location-based queries should be able to run without violating customer's privacy in order to prevent other businesses (small or big businesses such as banks, retail shops, etc) track MOs customers.

6) Session Management: This requirement is used to ensure that each session in big data application is handled properly based on the security requirements.

- **If sessions are used after authentication of users of big data applications:** Session identifier should be created on the server side and sent back to client at UI or Application Layer. During session duration, client side should include this identifier in its requests and the validity of it should be checked on the server side. (Note that due to mobility of users, frequent Internet Protocol (IP) address changes occur in cellular networks. Therefore, it is challenging to check session identifier created during as session with the IP address of the client.)
- **Synchronization:** Timestamps are essential part for session management. The system clocks should be synchronized through protocols such as Network Time Protocol (NTP).

7) Logging: This requirement is used to ensure that the logging of big data application inside MO is performed appropriately. For big data applications, the errors and access logs of applications across cluster need to be tracked. Moreover, distributed logging needs to performed in accordance with the policies of MOs as well as regulatory bodies.

B. Key security challenges in big data clusters

There are significant security challenges that need to be addressed before and after building a Hadoop cluster [21]. Some of these keys security aspects that are as follows:

1) Hadoop has rapid innovation cycle in an open source community which makes it difficult to devise a stable security procedure.
2) Multiple of feeds of information coming into the MO's data center in real-time have different protection needs and need to be handled in seperate case.
3) Due to distributed existence of data inside cluster, it is difficult to provide security against unauthorized or unauthenticated access. For example, rogue tasks and task trackers can impersonate real services.
4) Due to existence of multiple copies of data, the administration security complexities arise.
5) During inter-communication between nodes of cluster, there are quite a lot RPCs which are vulnerable for *man-in-the-middle* attacks. Appropriate encryption of data in transit needs to be devised while not compromising from performance including job duration and workload enhancements.
6) Due to the nature of the cluster formation, it is difficult to define a secure gateway where a DMZ or firewall can be set up. Therefore, it is still a difficult to find the best networking design approach for protection of the Hadoop cluster.
7) Frameworks such as *Zookeeper* or *YARN* manage node coordination and access control in order to detect failure of nodes, but not specifically designed for ensuring security of enterprise/MO's applications.
8) Due to the nature of Not-Only SQL (NoSQL) databases where referential integrity and data validations are absent, the data validations of big data applications must be performed at Hadoop Web UI Layer.
9) There exist issues with Hadoop integration with existing enterprise/MOs security services.

Therefore, due to various challenges involved as listed above, the security approach that MOs should develop must be holistic and should be able to conform to the various requirements specific to MOs. In the next section, we demonstrate the results of running vulnerability tests against the considered big data application and discuss some of the key observations with regard to previously mentioned requirements.

IV. Vulnerability Tests Against Big Data Application

In order to study security flaws of the example big data application platform deployed inside the MO, we have run extensive vulnerability tests for the architecture given in Section II. Some of the key findings are as follows:

- The SSL should always be used while the passwords are being transmitted via HTTP. All authentication process should be made over encrypted connections. If a user transport guarantee of integral or confidential is declared, all user name and password information will be sent over a secure connection using HTTP over SSL (HTTPS). The SSL protocol is an Internet standard, often used to provide secure access to Web sites, using a combination of public key technology and secret key technology. Secret key encryption (also called symmetric encryption) is faster, but asymmetric public key encryption provides for better authentication, so SSL is designed to benefit from the advantages of both. SSL authentication is based on digital certificates that allow Web servers and clients to verify each others identities before they establish a connection. This is called mutual authentication.
- User identification via brute force attack in password reminder section can be high where where CAPTCHA should be used for every request. CAPTCHAs are, by definition, fully automated, requiring little human maintenance or intervention to administer. This has obvious benefits in cost and reliability. CAPTCHA provides additional security against brute-force attacks on the web application. Brute-force attack monitoring should be enabled on the Web application by default. During the time that the application believes that a brute-force attack is occurring, it requires all users to specify the CAPTCHA response when logging in to the web application. CAPTCHAs based on reading text or other visual-perception tasks prevent blind or visually impaired users from accessing the protected resource. However, CAPTCHAs do not have to be visual. Any hard artificial intelligence problem, such as speech recognition, can be used as the basis of a CAPTCHA. Some implementations of CAPTCHAs permit users to opt for an audio CAPTCHA. Other implementations do not require users to enter text, instead asking the user to pick images with common themes from a random selection.

- Injecting client side scripts into web pages through Cross-Site Scripting (XSS) type of vulnerability can be commonly found in web applications utilizing big data in back-end. This violates one of the requirements of big data application which requires encrypted HTTP traffic in Hadoop Web UI. XSS refers to client-side code injection attack wherein an attacker can execute malicious scripts (also commonly referred to as a malicious payload) into a legitimate website or web application. XSS is amongst the most rampant of web application vulnerabilities and occurs when a web application makes use of unvalidated or unencoded user input within the output it generates. By leveraging XSS, an attacker does not target a victim directly. Instead, an attacker would exploit a vulnerability within a website or web application that the victim would visit, essentially using the vulnerable website as a vehicle to deliver a malicious script to the victims browser.
- Denial-of-service attacks through given links (e.g. send new password) can be performed while limiting the user' access into the web application through successive password reset trials. Two-factor authentication is a 'strong authentication' method, as it adds another layer of security to the password reset process. In most cases this consists of Preference Based Authentication plus a second form of physical authentication (using something the user possesses, i.e. Smartcards, USB tokens, etc.). One popular method is through SMS and email. Advanced Self Service Password Reset (SSPR) software requires the user to provide a mobile phone number or personal e-mail address during setup. In the event of a password reset, a PIN code will be sent to the user's phone or email and they will need to enter this code during the password reset process.

V. Conclusions and Future Work

Next generation cellular network infrastructures envisioned for 5G will need to cope with huge amounts of data traffic going through an operator's network infrastructure. In order to increase their share as well as revenue in the market, MOs should find new value-added ways to exploit this information and collaborate with other vertical market players on a common built platform deployed inside MO's premises. This big data exploitation will of course bring additional security challenges to MOs which need to be tackled specifically. In this paper, we have described big data security issues where requirements, challenges and preservation of private data in big data applications inside MO are investigated in details. Moreover, we have run some vulnerability tests against an example big data application deployed inside MO's premises. The vulnerability results show some of the key findings and map some of the missing requirements into the investigated big data application scenario. As a future work, more detailed analysis of integration challenges of 5G networks with evolving open source big data community (i.e. with Hadoop ecosystem) can be investigated.

References

[1] A. McAfee, E. Brynjolfsson, T. H. Davenport, D. Patil, and D. Barton, "Big data," *The management revolution. Harvard Bus Rev*, vol. 90, no. 10, pp. 61–67, 2012.

[2] M. Chen, S. Mao, and Y. Liu, "Big data: A survey," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 171–209, 2014.

[3] GSMA, "GSMA, The Mobile Economy 2015," *White Paper*, 2015.

[4] C. D. G. Romero, J. K. D. Barriga, and J. I. R. Molano, "Big data meaning in the architecture of iot for smart cities," in *International Conference on Data Mining and Big Data*, pp. 457–465, Springer, 2016.

[5] G. Suciu, V. Suciu, A. Martian, R. Craciunescu, A. Vulpe, I. Marcu, S. Halunga, and O. Fratu, "Big data, internet of things and cloud convergence–an architecture for secure e-health applications," *Journal of medical systems*, vol. 39, no. 11, p. 141, 2015.

[6] D. Das, O. OMalley, S. Radia, and K. Zhang, "Adding security to apache hadoop," *Hortonworks, IBM*, 2011.

[7] "Hadoop in secure mode." https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/SecureMode.html, 2015. [Online; accessed 04-February-2017].

[8] A. A. Cardenas, P. K. Manadhata, and S. P. Rajan, "Big data analytics for security," *IEEE Security and Privacy*, no. 6, pp. 74–76, 2013.

[9] R. Lu, H. Zhu, X. Liu, J. Liu, and J. Shao, "Toward efficient and privacy-preserving computing in big data era," *Network, IEEE*, vol. 28, pp. 46–50, July 2014.

[10] P. P. Sharma and C. P. Navdeti, "Securing big data hadoop: a review of security issues, threats and solution," *Int. J. Comput. Sci. Inf. Technol*, vol. 5, 2014.

[11] C. Mora *et al.*, "Top ten big data security and privacy challenges," *Cloud Security Alliance*, 2012.

[12] A. Cuzzocrea, "Privacy and security of big data: Current challenges and future research perspectives," in *Proceedings of the First International Workshop on Privacy and Secuirty of Big Data*, PSBD '14, (New York, NY, USA), pp. 45–47, ACM, 2014.

[13] Zettaset, "The big data security gap: Protecting the hadoop cluster," April 2014.

[14] "Apache sentry (incubating)." http://sentry.apache.org/, 2016. [Online; accessed 04-February-2017].

[15] "Project rhino." https://github.com/intel-hadoop/project-rhino/, 2015. [Online; accessed 04-February-2017].

[16] "Apache accumulo." https://accumulo.apache.org/, 2015. [Online; accessed 04-February-2017.

[17] "Datastax." http://www.datastax.com/products/datastax-enterprise, 2015. [Online; accessed 04-February-2017].

[18] "Dataguise." http://www.dataguise.com/, 2015. [Online; accessed 04-February-2017].

[19] C. Neuman, S. Hartman, T. Yu, and K. Raeburn, "The kerberos network authentication service (v5), RFC 4120," July 2005.

[20] J. Linn, "The kerberos version 5 GSS-API mechanism." http://www.ietf.org/rfc/rfc1964.txt/, RFC 1964, 1996.

[21] N. Sawant and H. Shah, *Big Data Application Architecture Q&A: A Problem-Solution Approach*. Apress, 2013.