Graph Database Security: Blockchain Solution and Open Challenges

Mohammad Shkoukani, Ahmad Mousa Altamimi

Department of Computer Science, Applied Science Private University, Amman, Jordan.
Email: m.shkokani@asu.edu.jo; a_altamimi@asu.edu.jo

***Abstract* - NoSQL graph database uses graph structures with nodes and edges to represent and store data. While the graph database was created to address the limitations of the existing relational databases, it was initially designed by not considering security as an essential feature. One can consider the analytic attack where data is queried and analyzed. Therefore, the graph database provides poor privacy and security protection. In this paper, we survey the major security issues for graph databases and outline open challenges. Specifically, the popular security attacks and threats were reviewed and categorized along with their state-of-the-art solutions. To gap the graph security requirements, a security model based on Blockchain technology is proposed. The proposed model supports the development of graph-based applications that preserve data security and integrity. It can be adapted as a stand-alone system or integrated with other systems, which enables applications to harness secured graph databases for many modern-day use applications and eliminates restrictions imposed by database models. We believe that this research forms a basis for broader future studies of using Blockchain technology to facilitate the development of different secure graph applications.**

Keywords - NoSQL Graph Database; Security; Blockchain Technology; Smart Contract; Ethereum

I. INTRODUCTION

The recent advance in cloud computing and distributed applications motivate the various adopting types of non-relational databases, known as NoSQL databases [1]. NoSQL databases have different types, each of which employs a specialized technique. One can consider here, the wide-column stores, document stores, key-value stores, and graph databases. Their primary advantage is to store unstructured data such as documents, multimedia, or social media efficiently. The graph database is explicitly designed to easily accommodate a huge amount of data efficiently, which is produced from the recent advance in cloud computing and distributed applications [2].

The graph database model is comprised of nodes and edges to represent entities and relationships. While the nodes represent physical or logical entities such as a person, place, or piece of data, the edges represent the relationship between these nodes [3]. A graph database is helpful as they show the relationships between relevant data. So, it can be found in many modern-day applications that comprised of multiple systems and process highly connected data such as social media and medical platforms [4]. Although these systems allow highly retrieved connected data, however, they lack common standards like synchronization, correctness, consistency, and cybersecurity [2].

Because graph databases are dynamic data portrayal with frequent data changes, the correctness and consistency are key challenges. For example, imagine a graph database used shown in figure 1. If the graph is undergoing an update, which creates (n4, n5) and deletes (n3, n4) concurrently. A path starting from host n1 to host n5 may erroneously be returned, even though no such path ever existed.

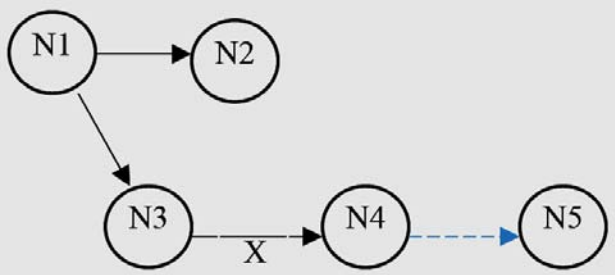

Figure 1. Graph database correctness and consistency problem

Security and privacy are other challenges for graph databases. Without dedicated attention paid to cybersecurity, technology is vulnerable to unauthorized access. According to [5], the NoSQL database was not designed initially by considering the security aspect as an important feature. Therefore, protecting these databases is the sole responsibility of NoSQL consumers by developing third-party tools and services. However, this may result in posing many threats to the collected data. One can consider, for example, the latest major cybersecurity massive breach that was revealed by Equifax (a US-based consumer credit agency) in financial health data [6]. The breach affected the data of 143 million U.S. consumers. The consumers' names, SSN, birth dates, and addresses were stolen and used to gain access to additional information such as medical records and accounts.

In short, this breach considered as one of the worst in terms of the number of people affected and the type of information released. To prevent such breach, it is required to capture, store, analyze and potentially share vast amounts of rapidly evolving information. Consequently, ensuring data

security (protecting sensitive data) in graph databases will be very costly and inefficient [7].

Furthermore, because the graph database stores large data sets, an additional layer of complexity is added to administrate such data. This is particularly because of the unique characteristics of graph queries, which makes the consistent queries are challenging for a graph. For example, traversals query such as reading a large portion of the graph takes a long time to execute. For instance, in the Facebook network, the average degree of separation is 3.5 [8]. This means that implementing a breadth-first traversal for traversing four hops results in reading all 1.59 billion users. One proposed solution for this problem is employing optimistic concurrency control or distributed two-phase locking techniques. However, poor throughput resulted when concurrent queries try to read large subsets of the graph [9].

Besides that, most popular state-of-the-art graph databases such as Neo4j [10] and Titan [11] employ heavyweight coordination techniques. Weakly consistent online graph databases [12-13] forgo strong semantics for performance, which limits their scope to applications with loose consistency needs and requires complicated client logic. Offline graph processing systems [14-15] do not permit updates to the graph while processing queries. Lightweight techniques for modifying and querying a distributed graph with strong consistency guarantees have proved elusive thus far.

Since there are notable issues of using the standard graph database, which may result in catastrophic consequences. It appears the demand for alternative technological solutions [16-17]. Blockchain is one of the most promising alternative new technologies within the field in recent years. Blockchain is not only an alternative supplement to the existing storing solutions, but also it is a new way of retrieving and transferring data transparency [18].

Blockchain was invented by Satoshi Nakamoto in 2009 as an online currency named Bitcoin [19]. It based on open distributed hash ledger between peer-to-peer nodes, the Bitcoin technology is a decentralized alternative to traditional monetary, and it eventually turned into what so-called technological revolution. Blockchain has been utilized in many different applications and not limited to monetary activities. For instance, it is employed in business, finance, production, import, export, industry, and many others, to secure information storage and improve information sharing and networking. With the help of this advanced technology, the gap in credentialing, efficient communication, and storing is filled up. Moreover, the system processes will become faster, easier, and safer [20].

Our focus on this work will be on the adoption of Blockchain technology in the graph database, where there have been little efforts spends on this new technology in such an environment. The work aims to develop a security model for NoSQL Graph-Oriented Databases based on the using of Blockchain technology. The goal of the model is to support the development of graph-based applications and to adopt it among corporations of this technology for preserving the integrity of stored data and protecting them from non-authorized access. This enables applications to harness secured graph databases for many modern-day use applications. We believe that this research forms a basis for broader future studies on using Blockchain technology to facilitate the development of different applications with data security and eliminate the need for third party systems.

The remaining paper is divided into five sections. Section 2 provides background material needed to understand the research problem and its significance. Section 3 provides a relevant literature review. The research's objectives and methodology are then specified in Section 4. The theoretical research model is outlined in Section 5, where the discussion is provided in Section 6. The conclusion and future work are finally given in Section 7.

II. BACKGROUND MATERIAL

In this section, the primary concepts that we need to be familiar with are introduced. Firstly, the graph database is briefly described. Then, some of the graph database use cases are discussed. After that, the graph database privacy and security are discussed. Finally, Blockchain development is given.

A. Graph Database

The importance of storing and investigating data in the form of the graph has been increasing [21-22]. The graph database utilizes graph structures for storing data. Each graph consists of two elements: a node that represents an entity (object, person, thing, etc.) and a relationship that represents the association between two nodes [3]. Recently, there are many systems that use a graph database, such as social networks, biological network, and web graph. Twitter can be considered as a good example of a graph database [23]. Figure 2 shows a simple graph data model for a pizza shop. For example, "pizza" and "food" nodes would have the relationship "is a kind of" pointing from "pizza" to "food". Here, each node may have one or more attributes.

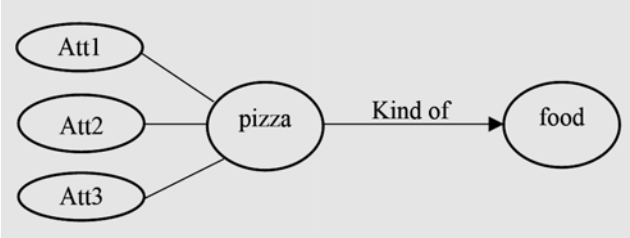

Figure 2. A simple Graph Data Model

In fact, there are many different graph database models [24], such as:

AllegroGraph is one of the originators in the current generation of graph databases, which provides special features for social network analysis.

DEX provides a Java library for the management of persistent and temporary graphs. It is oriented to ensure a good performance in the management of very large graphs.

HyperGraphDB is a database that implements the hypergraph data model where the notion of the edge is extended to connect more than two nodes. It is useful for modelling data of areas like artificial intelligence and bioinformatics.

InfiniteGraph is a database-oriented to support largescale graphs in a distributed environment. It focuses on extend business, social and government intelligence with graph analysis.

Neo4j is an open-source graph database that is implemented in Java and developed by Neo technology. It uses graph structure with nodes, edges and properties to store data.

Sones is a graph database, which provides inherent support for high-level data abstraction concepts for graphs. It defines its own graph query language and an underlying distributed file system.

B. Graph Database Use Cases

Financial applications: The Relational Database Management System (RDBMS) is initially designed to store highly expensive data and has proven to be stable and effective in transaction processing environments. For example, the RDBMS excels in large-scale banking data (credit-card transaction processing and cyclic billing operations). However, RDBMS are not capable of exploring relationships to uncover crimes like forensic analysis or identity theft. However, the emergence of the NoSQL database like graph databases represents an alternative with far less overhead and performance penalties [23]. It can navigate connections in real-time to discover patterns and stop fraud.

Biotech applications: Because graph database is capable of storing multimodal databases, object databases, cloud databases, and graph databases. It is utilizing by governments and companies for creating innovative applications to study diseases and discover new treatments [23].

Online retail applications: Graph databases are used to exploit the graphs quickly and find links between data. This results in maximizing the efficiency for providing a fastest and efficient method to handle online transactions or financial data. Moreover, Graph databases are minimizing the overhead for searching for data [2].

C. Graph Database Security

The main goal of NoSQL graph-oriented databases is to handle a massive amount of data. Some of this data may be personal or sensitive information and need to be protected. However, according to [5], NoSQL databases were initially not designed by considering security as an important feature. They provide poor privacy and security protection and depend on third-party tools and services to protect the stored data [25]. Moreover, because of graph databases offer great promise for resolving complex, interrelated relations, attackers, can exploit such relations to rob individuals of their privacy.

One proposed solution for this problem is employing optimistic concurrency control or distributed two-phase locking techniques. However, poor throughput resulted when concurrent queries try to read large subsets of the graph [13]. Another solution is to use the relational databases security mechanisms like [16-17], [26]. However, graph databases fundamentally differ from relational databases. For example, relational databases cannot visualize relationships between data abstract, as the data is stored in rows and columns. By contrast, graph databases enable the visual representation of the relationships between data points using nodes, edges and properties. So, many key security features in relational databases have been left in non-relational management systems (e.g., access control) and the responsibility is given to a third party to develop security services or tools, which raise many security breaches.

Since there are notable issues of using standard security mechanisms, which may result in catastrophic consequences, it appears the demand for alternative technological solutions. In this paper, a security model is developed for NoSQL Graph-Oriented Databases based on the using of Blockchain technology.

D. Blockchain Development

A blockchain is a type of electronic ledger, distributed database that forms a new way of documenting data on the internet [19]. The data recorded on a blockchain can take any form, transactions, identities, an agreement between two parties, or any data that store in blocks. Blocks are appended and linked together, forming a chain of blocks through a process called hashing as shown in Figure 3.

A hash function is a cryptographic algorithm that takes any arbitrary input size and produces an output of fixed length size. The hash is produced by running the selected cryptographic hash algorithm on the blockchain content (i.e., block content) to generate the unique hash value of the block content, the hash should be easily produced for any arbitrary input, but it should very hard to retrieve back the input based on the output hash. Any changes in the original input data should result in extensive and uncorrelated changes to the output hash [27]. Moreover, the Blockchain is considered so trustworthy, the recorded data in blocks cannot be altered without having to modify all the blocks after it [28]. Additionally, Blockchain is not controlled by anyone, and it is duplicated in its entirety across participants. This has a

great advantage, where a user can view the chain from anywhere if he has the right crypto keys.

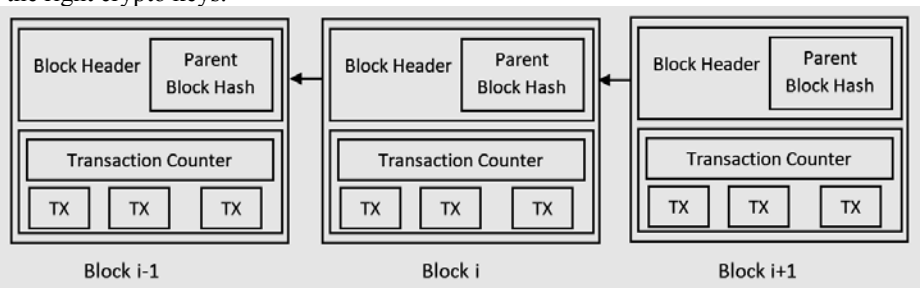

Figure 3. A simple Blockchain Architecture

Finally, today, developers and researchers in various fields believe that applications of Blockchain are not limited to cryptocurrencies, but it extends to cover most of the known traditional and nontraditional implantations as medicine, education, industry, fabrication, distribution and many others, nowadays variety sectors are looking for ways to integrate Blockchain into their infrastructures.

III. RELATED WORK

There are many works proposed to analyze the graph vulnerabilities and attacks. Most of them emphasized on analyzing the vulnerability and proposing and reviewing new techniques to prevent them. Noel et al. [29] proposed a method for modelling, analyzing, and furthermore visualizing attack graphs. Their technique can associate attack paths with security events. The analyzed attack graphs are stored in Neo4j; after that, they queried and analyzed the attack relationships, and visualized queries results.

On the other hand, other researchers proposed novel methods. Tao et al. [30] proposed a graph database-based hierarchical multi-domain network security situation awareness data storage method. In the proposed model, they used graph database Neo4j to store data, which can query the internal and external attacks conveniently, in addition, to query the basic network information easily. Shangqi Lai et al. [31] proposed an encrypted graph database, which enabled privacy-preserving rich queries in the context of social network services such as Facebook. The proposed system provides an encrypted structural data model to facilitate parallel and encrypted graph data access by generating an encrypted index on a distributed graph model.

Moreover, the design phase was also considered in many works. Boza et al. [32] reviewed the design of Neo4J and OrientDB, which are the most widespread graph databases, identifying a few security problems, improper default configurations and leaks. Barik et al. [33] used a constrained graph model to examine network vulnerability, and they proposed an extended attribute graph model-based graph constraint specification language, which will be used to examine the attack graph-based network vulnerability. By implementing the attribute graph model in Neo4j, they verified that using these constraints could guarantee the accuracy of the attack graph generation and analysis process.

Regarding Neo4J, they recognized that bad configuration habits of the database, making them vulnerable on the web. In regard to OrientDB, they recognized there are also security problems due to the default configuration. According to the security problems which were discovered force to release a new version of the analyzed graph databases, Neo4j and OrientDB.

However, these works are differing from our work as we are integrating new leading-edge technologies (the Blockchain) to protect the stored data from attacks. More details about this integration are presented in Section 5.

IV. METHODOLOGY AND RESEARCH OBJECTIVES

Companies collect our data and keep the promise of keeping them safe. However, the collected data stored in graph databases are being attacked or being hacked by malicious third parties. According to the Verizon data breach investigation report, the year of 2018 registered 53k security incidents and more than 2k data breaches in 65 different countries [6]. As a classic example: The Facebook platform has not only suffered major data breaches but has also sold data to its partners without explicit user consent. This is obviously causing a problem for both users and companies implicitly charged with fines and penalties.

Thus, the main objective of this work is to develop a model that integrates the Blockchain technology to improve the security of graph databases. The Blockchain bases system provide cryptographic algorithm (Hash algorithm), which plays an important role in assuring data security. Furthermore, all the linked data can be shared based on certain requirements (Smart contract), which ensures the safety and privacy of the stored data [34].

Moreover, through such integration, applications can harness graph databases with transactional data for many modern-day use applications. As a result, production environments are not impacted while enterprises transition to modern-day applications. It also eliminates restrictions imposed by data models and database vendors by harnessing graph data relationships with document metadata for better security and privacy.

V. THE RESEARCH MODEL

The main goal of graph databases is to store a large amount of data, were not designed with solid security features, and their protection depends on where some of this data is sensitive and need to be protected. However, such databases on third-party tools and services [25]. However, recent technology can bridge the security gap and provide the necessary security mechanisms.

Blockchain, for example, stores data of any form (transactions, identities, or any agreement between two parties) in block structures. Each block is appended and linked to another block, forming a chain of blocks through a process called hashing. A hash function is a cryptographic algorithm that takes any arbitrary input size and produces an output of fixed length size. Through, all the linked data can be shared based on certain rules (Smart contract), which ensures the safety and privacy of the stored data [34].

Moreover, the Blockchain is considered so trustworthy because the recorded data in blocks cannot be altered without having to modify all the blocks after it [28].

Now, recall that the graph database consists of two elements: a node, which represents an entity (object, person, thing, etc.) and a relationship that represents the association between two nodes. Twitter can be considered as a very good example of a graph database. Figure 4 shows a graph data model for simple Twitter users, which has three nodes/users, Billy, Harry, and Tom and the relationship follows that describes how each user is connected to another one.

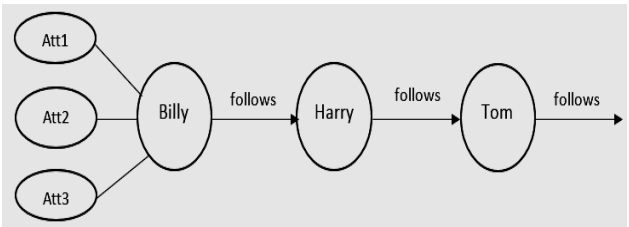

Figure 4. A Simple Twitter Graph Data Model

Here, the idea is to create graphs as linked blocks in order to implement the required security countermeasures. The process starts with the first block of a Blockchain, which is called the genesis block and represents the first node in the graph. A block consists of the block header and the block's body, as shown in figure 5.

In particular, the block header includes a block version, which indicates which set of block validation rules to follow. Merkle tree root hash, which is the hash value of all the data in the block and the relationship with the other node. The timestamp, Nonce, and Parent block hash, which is a 256-bit hash value that points to the previous block. On the other hand, the block body is composed of a counter and the stored data along with its attributes.

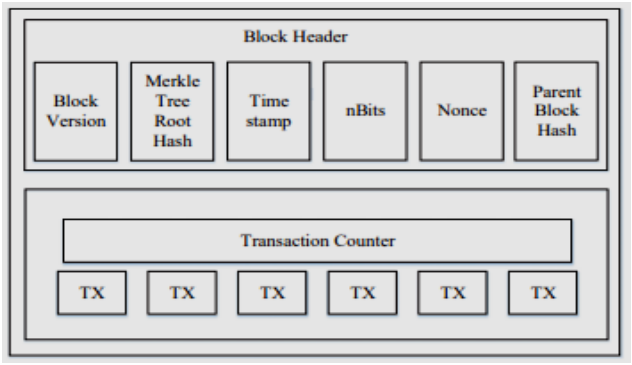

Figure 5. The Standard Block Structure

The chain is then growing as new blocks or nodes are appended to it continuously. In order to represent the relationship, we will depend on the links that connect the nodes together. Specifically, in Blockchain, each block is linked with a previous block through the hash contained in the block header.

The hash is produced by running the selected cryptographic hash algorithm on the value of all the data in the block and the relationship to generate the unique hash value. This value will be placed in the next node that participates with it with the relationship. Figure 6 illustrates the corresponding Blockchain.

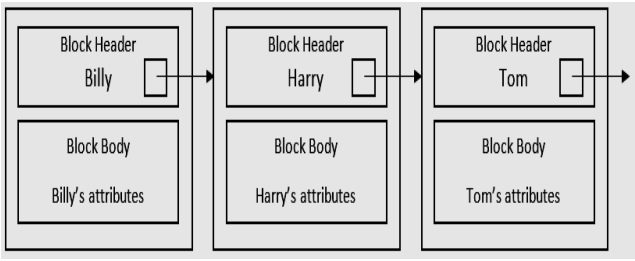

Figure 6. The Corresponding Twitter Blockchain

The above-depicted illustration is a simplistic representation of a blockchain-based system specifically designed for storing and sharing data stored in graph databases. The model can be designed using any type of Blockchain technologies (Public or Private Blockchain). In Public Blockchain, the chain is opened to anyone to participate and see all the stored data. This can be helpful in applications with no sensitive or private data. In contrast, the Private Blockchain has permissions to join the chain, and the stored data can be seen by only the authorized users [28].

As proof of our concept, the model can be implemented using Ethereum. Ethereum is an open-source distributed platform that facilitates the creation of Blockchain and its rules in the form of complex if-then statements [35]. Using Ethereum, one can manage the infrastructure in order to authenticate the users.

VI. DISCUSSION

Graph database provides poor privacy and security protection because it was not designed by considering security as an essential feature. This results in security breaches and threats. For example, since the data is being continuously received through a variety of platforms and placed in an information management system. This places all data under an analytic attack where data can be queried, but the system only tracks data points. The recent data breach from Target was a very methodical campaign with multiple steps that took place over a month. The entry point took place through one of Target's contractors, which received a security alert at the earliest stage of the attack; however, identified it as a false alarm.

In this paper, we started by surveying the major security issues for graph databases and outline some open challenges. Specifically, the popular security attacks and threats were reviewed and categorized along with their state-of-the-art solutions. Then, a security model based on Blockchain technology is proposed to gap the graph security requirements particularly the analytic attack. This is achieved because of the unique Blockchain characteristics: firstly, by employing the private Blockchain that has permissions to join the chain, and the authorized users can only see the stored data. Secondly, by utilizing the Blockchain cryptographic algorithm (Hash algorithm), which plays an important role in assuring data security. Finally, by ensuring the Blockchain smart contracts, where all the linked data are shared based on certain requirements, which ensures the safety and privacy of the stored data.

The proposed model can be easily implemented using Ethereum (an open-source distributed platform that facilitates the creation of Blockchain and its rules). Using Ethereum, one can manage the Blockchain infrastructure in order to authenticate the users. So, the implemented model can be adapted as a stand-alone system or integrated with other systems. Such adaption will facilitate the development of secured graph-based applications and eliminate the restrictions imposed by database models or database vendors. As a result, production environments will not be impacted while enterprises transition to modern-day applications. We believe that more works can based on this research for using Blockchain technology to facilitate the development of different secure graph applications.

VII. CONCLUSION AND FUTURE WORK

This paper starts with an overview of the graph databases and blockchain technology and their related concepts that resulted in the generation of motivational and analyzing scenario. The key research points were focused on the security aspects of the graph databases and explained how threat Graph provides analysts and integrators with real-time, forensic-level visibility into all endpoint activity, which presents privacy and security concerns.

Then the paper presented a security model to solve the stated problem. The model is specifically designed to enhance the graph database security by integrating the Blockchain technology. Through such integration, applications can harness graph databases with transactional data for many modern-day use applications. The proposed model can be adapted as a stand-alone system. But it's also equipped to make integration with multiple systems easier.

As future work, we are planning to implement our model on the different types of NoSQL databases (key-value databases, column-oriented databases, document-oriented databases) and then compare the resulted security level with the one got from graph database that will be used in the research.

ACKNOWLEDGEMENT

The authors are grateful to the Applied Science Private University, Amman, Jordan, for the full financial support granted to this research.

REFERENCES

[1] Meier, Andreas and Kaufmann, Michael, "Nosql databases", Springer, pages: 201-218, 2019.

[2] Nayak, Ameya and Poriya, Anil and Poojary, Dikshay, "Type of NOSQL databases and its comparison with relational databases", International Journal of Applied Information Systems, Vol. 5, No. 4, 2013, pp. 16-19.

[3] Angles, Renzo," The Property Graph Database Model", Springer, 2018.

[4] Shah, Faaiz and Castelltort, Arnaud and Laurent, Anne, "Handling missing values for mining gradual patterns from NoSQL graph databases", Future Generation Computer Systems, 2019.

[5] Ahmad, Khaleel and Alam, Mohammad S and Udzir, Nur Izura, "Security of NoSQL database against intruders", Recent Patents on Engineering, Vol. 13, No. 1, 2019, pp. 5-12.

[6] Novak, Alison N and Vilceanu, M Olguta, "The internet is not pleased": twitter and the 2017 Equifax data breach", The Communication Review, Vol. 22, No. 3, 2019, pp. 196-221.

[7] Angles, Renzo and Gutierrez, Claudio, "Survey of graph database models", ACM Computing Surveys (CSUR), Vol. 40, No. 1, 2008, pp. 1-39.

[8] Three and a half degrees of separation, https://research.facebook.com/blog/three-and-a-halfdegrees-of-separation, Access date 2020.

[9] Dubey, Ayush and Hill, Greg D and Escriva, Robert and Sirer, Emin G, "Weaver: a high-performance, transactional graph database based on refinable timestamps", arXiv preprint , 2015.

[10] Neo4j. http://neo4j.org, Access date 2020.

[11] Titan. https://github.com/thinkaurelius/titan/wiki, Access date 2020

[12] Bronson, Nathan and Amsden, Zach and Cabrera, George and Chakka, Prasad and Dimov, Peter and Ding, Hui and Ferris, Jack and Giardullo, Anthony and Kulkarni, Sachin and Li, Harry, "Facebook's distributed data store for the social graph, 2013", pp. 49-60.

[13] Cheng, Raymond and Hong, Ji and Kyrola, Aapo and Miao, Youshan and Weng, Xuetian and Wu, Ming and Yang, Fan and Zhou, Lidong and Zhao, Feng and Chen, Enhong, "Kineograph: taking the pulse of a fast-changing and connected world", Proceedings of the 7th ACM european conference on Computer Systems, 2012, pp. 85-98.

[14] Gonzalez, Joseph E and Low, Yucheng and Gu, Haijie and Bickson, Danny and Guestrin, Carlos, "Powergraph: Distributed graph-parallel

computation on natural graphs", Presented as part of the 10th proceeding Symposium on Operating Systems Design and Implementation, 2012, pp. 17- 30.

[15] Salihoglu, Semih and Widom, Jennifer, "Gps: A graph processing system, Proceedings of the 25th International Conference on Scientific and Statistical Database Management, 2013, pp. 1-12.

[16] Eavis, Todd and Altamimi, Ahmad, "OLAP authentication and authorization via query re-writing", DBKDA 2012, pp. 137, 2012.

[17] Shkoukani, Mohammad and Altamimi, Ahmad Mousa and Qattous, Hazem. "An Experimental Study to Evaluate the Integration of Various Security Approaches to Secure Transferable Data," International Journal of Simulation Systems, Science & Technology, Vol. 20, No. 1, 2019.

[18] Puthal, Deepak and Malik, Nisha and Mohanty, Saraju P and Kougianos, Elias and Yang, Chi. "The blockchain as a decentralized security framework [future directions]," IEEE Consumer Electronics Magazine, Vol. 7, No. 2, 2018, pp. 18-21.

[19] Champagne, Phil. "The book of Satoshi," Lexington, KY: e53 Publishing, 2014.

[20] Bhargava, Richa., "Blockchain Technology and Its Application: A Review", IUP Journal of Information Technology, Vol. 15, No. 1, 2019, pp. 7-15.

[21] Medhi, Surajit and Baruah, Hemanta K., "Blockchain Relational database and graph database: A comparative analysis", Journal of Process Management. New Technologies, Vol. 5, No. 2, 2017, pp. 1-9.

[22] Martnez Porras, Alexandra and Mora Rodr, "A Comparison between a Relational Database and a Graph Database in the context of a Personalized Cancer Treatment Application", 2016.

[23] Robinson, Ian and Webber, Jim and Eifrem, Emil, "Graph databases", O'Reilly Media, 2013.

[24] Angles, Renzo, "A comparison of current graph database models", IEEE 28th International Conference on Data Engineering Workshops, 2012, pp. 171-177.

[25] Zahid, Anam and Masood, Rahat and Shibli, Muhammad Awais, "Security of sharded NoSQL databases: A comparative analysis", IEEE Conference on Information Assurance and Cyber Security (CIACS, 2014), pp. 1-8.

[26] Altamimi, Ahmad and Eavis, Todd, "Securing Access to Data in Business Intelligence Domains", International Journal on Advances in Security, Vol. 5, No. 3, 2012.

[27] Anjum, Ashiq and Sporny, Manu and Sill, Alan, "Blockchain standards for compliance and trust",IEEE Cloud Computing, Vol. 4, No. 4, 2017, pp. 84-90.

[28] Nofer, Michael and Gomber, Peter and Hinz, Oliver and Schiereck, Dirk, "Blockchain", Business and Information Systems Engineering, Vol. 59, No. 3, 2017, pp. 183-187.

[29] Noel, Steven and Harley, Eric and Tam, Kam Him and Gyor, Greg, "Big-data architecture for cyber attack graphs representing security relationships in nosql graph databases ",Citeseer, 2015.

[30] Tao, Xiaoling and Liu, Yang and Zhao, Feng and Yang, Changsong and Wang, Yong, "Graph database-based network security situation awareness data storage method", EURASIP Journal on Wireless Communications and Networking, Vol. 2018, No. 1, 2018, pp. 294.

[31] Lai, Shangqi and Yuan, Xingliang and Sun, Shi-Feng and Liu,, "GraphSE: An Encrypted Graph Database for Privacy-Preserving Social Search", Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security, 2019, pp. 41-54.

[32] Boza, M. and Muñoz, A., "(In) Security in Graph Databases - Analysis and Data Leaks", Proceedings of the 14th International Joint Conference on e-Business and Telecommunications (ICETE, 2017), pp. 303-310.

[33] M.S. Barik, C. Mazumdar, A. Gupta, "Network vulnerability analysis using a constrained graph data model", International Conference on Information Systems security, 2016, pp. 263-282.

[34] Maesa, Damiano Di Francesco and Mori, Paolo and Ricci, Laura, "A blockchain based approach for the definition of auditable Access Control systems", Computers and Security, Vol. 84, 2019, pp. 93-119.

[35] Swan, Melanie, "Blockchain: Blueprint for a new economy", O'Reilly Media Inc., 2015.