**IS 651: Distributed Systems**

**Final Project Report**

**Type1: Literature Review**

**Topic: Security Issues in Distributed Databases**

**Instructor: Dr Sisi Duan**

**University Of Maryland Baltimore County**

**Submitted By**

|  |  |
| --- | --- |
| **Dharmil Shah** | **SN38896** |
| **Krishna Patel** | **YA04014** |
| **Arjun Kumar** | **MC31132** |

1. **Introduction**

The security techniques to be used in distributed databases are the important concerns. Several elements of distributed databases security are identified, like authentication, authorization, encryption and system protection. In initial days, the security management environment was based on single authority systems but now the focus is on the development of per activity, authorities and groups with shared responsibilities. The general security attacks on the distributed systems are eavesdropping (gaining secret information), masquerading (making assumptions on the identity of users), and message tempering (changing the content of the message), replaying the message and denial of services. The trustworthiness of distributed systems is important in several environments. For expressive economy the term security is used to represent both its traditional meaning as well as those notions carried by the term privacy. Before discussing the factors affecting security in distributed databases, an overview of distributed databases architecture is presented and used as a framework for subsequent analysis.

**1.1 Distributed Database System**

A distributed database system (DDBS) is a series of many logically connected databases that are geographically spread over a virtual network on separate machines (otherwise known as sites) All sites that engage in the spread database have local sovereignty in the sense that the database at each location has complete power over itself in terms of data management.The sites will even interoperate whenever needed. The operator of a distributed database has the illusion, the whole database is local, except for potential contact delays between the locations. That is since a centralized network is a collective union of all the pages, so the consumer is shielded from the delivery.

For various reasons, DDBS is preferred to a non-distributed or central Database System (DBS). Distribution within an enterprise is quite common. For eg, separate divisions of the company can be situated at specific geographic locations. Logical distinctions may also occur within a section such as department of human resources and department of administration. We should view every logical division as a site. In addition to having other benefits of a distributed network, such as redundancy and fault tolerance, a DDBS delivers the functionality of a central DBS to the device at each location. Though the data delivery principle is appealing, the data is very small. Data delivery also allows processes for managing competitiveness and recovery complex.

Distribution existence in a DDBS complicates the communication process of the locations. Three major problems differentiate the DDBS from the conventional DBS, which are:

* **Replication –** this may be achieved by placing several versions of a table in its entirety or table partitions on different pages. Replication delivers at least two advantages: high availability and better performance.
* **Fault tolerance –** issues during site collapse due to processing are complicated. Failures can occur at various levels of a DDBS. Categorized DDBS failures under four headings: transaction-central failures, website failures, media failures and network failures.
* **Concurrency Control-** In DDBS the transaction principle is complicated as a transaction can access data processed at more than one location. A trade is usually broken into subtransactions that can be performed at various locations. Transactions are sequences of actions done by a customer or program. The acts specified in a transaction are either completely implemented, or not implemented (known as atomicity). The confusion regarding the order in which sub-transactions are performed will result in different performance or different database state. Concurrency control is in this sense a process of controlling the relative order of two or more sub-transactions which interfere with each other.

**1.2 Need for Distributed Systems**

**Organizational and economic reasons:** Multiple organisations are not organized, so the organization's framework works more easily with a relational database method. Perhaps the most significant explanation for creating centralized systems are the operational and economic motives.

**Interconnection of existing databases:** Where there are many repositories in an enterprise and there is a need to execute global operations, centralized databases are the obvious response. The distributed database is built from local repositories that already exist. This method will involve consolidation, and far less work is needed than creating a new centralized database.

**Incremental Growth growth:** An enterprise expands by introducing additional operational structures including new offices, new facilities, etc., and then hierarchical ledger strategy encourages gradual development with limited effects on established structures.

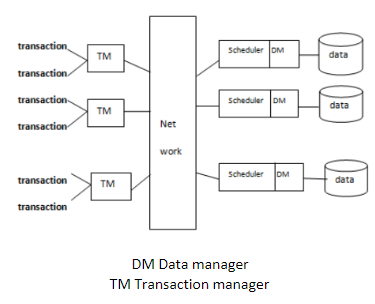
**Decreased overhead contact:** Multiple systems are local, raising overhead communication with respect to a centralized database.

**Performance:** The presence of multiple autonomous processors allows efficiency to improve. The benefit of distributed databases is that the decomposition of data represents domain-dependent parameters that optimize client location, while minimizing mutual interference among different processors. The load is spread between the different processors, and crucial bottlenecks are eliminated, such as contact network or basic utilities.This impact is a consequence of the autonomous processing capacity prerequisite for local applications specified in the distributed database concept.

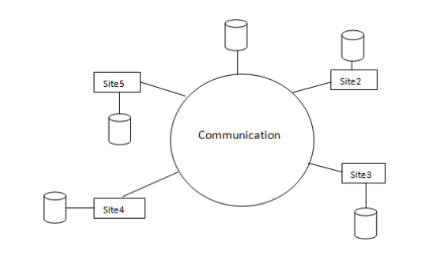
**Redundancy and availability:** For greater redundancy and availability, the distributed database solution can be used especially for redundant data.

**1.3 Architecture of Distributed database**

A distributed database management system (DDBMS) includes a networked array of pages. -- site operates one or more of the following program modules: a transaction manager(TM), a data manager(DM), and a scheduler or simply scheduler for competition regulation.



A site may act as a client, server, or both in a web-server model. A device runs only the module TM and a server runs only the application DM and Scheduler. Single server stores a portion of the database. Each data object can be stored on any computer, or on many servers, redundantly. Fig-1 displays the application server configuration device design.



Distributed Database Environment

Users communicate with the DDBMS by running web requests or computer programs that are transactions. TMs manage transaction-to-database contact. The TM that is considered the initiating TM at the location where the transaction began. The suggesting TM collects transactions provided by a client, then forwards them to the schedulers concerned. A scheduler's purpose is to organize operations in such a way that the resulting execution is right. By performing procedures, DMS controls the entire database and is liable for recovery from errors. Transactions interact with TMs that coordinate and handle data through schedulers, DMs. Architecturally a DDBS consists of a collection of database sites (possibly empty) and a non-empty set of data sites. The data sites have the potential to store data where the database sites do not. The above only operates the user interface to allow application access at web locations, as seen in above figure. Distributed database processing problems involve agreeing on a method to perform each question across the network in the most cost-effective manner. The response time and throughput involve two indicators of database efficiency. The response time is the time taken by a machine to address a question, and the total number of transactions that move through the machine successfully. (Quasim, Mohammad. (2014))

**2. Data Security Approaches**

This part talks about the different approaches used in the Project.

**2.1 Authentication**

Authentication is the process of recognizing a user’s identity. It is the mechanism of associating an incoming request with a set of identifying credentials. The credentials provided are compared to those on a file in a database of the authorized user’s information on a local operating system or within an authentication server.

The authentication process is the first of accessing the data from the database. The identity is provided in the form of a user ID. The security system will search all the abstract objects that it knows and find the specific one of which the actual user is currently applying. Authentication is adopted to ensure that only authentic users can use the database. To provide authentication digital certificates are used. Besides, login is restricted through username/password combination.

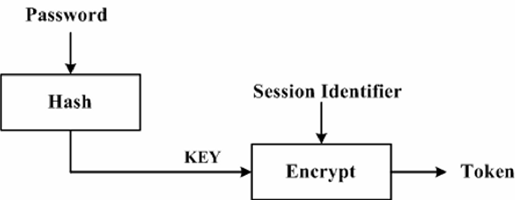
Authentication using cryptography techniques is more secure. It helps to encrypt and decrypt the message or input for authentication, it secures the communication of the user and the database while authenticating. This process, embedded in protocols and written in software that runs on operating systems and networked computer systems, involves public and private key generation of data encryption/decryption, digital signing and verification for message authentication, and key exchange. Information security and cryptography are interconnected and share the common services of protecting the confidentiality, integrity and availability of the information ignoring data form. The length and strength of the Cryptography keys are considered an important mechanism. The keys used for encryption and decryption must be strong to produce strong encryption. (Mohammad, Abdelfatah & Omaima, 2014)

Symmetric encryption uses a single private key which is used for both encryption and decryption.

Asymmetric encryption uses a secret key, which is known only to the user and public key disseminated widely.

The authentication process can be done successfully through the following three classical ways:

1. Something known: password, or personal identification number (PIN)
2. Something possessed: smart card or token
3. Something inherent: face (Biometric) detection and recognition, fingerprint, voice, retina or iris characteristics.



**2.1.1 Kerberos Authentication Protocol**

Kerberos is a trusted third party network authentication protocol providing security services over a network. The third part is the Kerberos security server that provides a network which can trust each other enabling mutual authentication between client and the database. It gives a solution to the problem of network security by providing tools of authentication and strong cryptography over the network to provide security to the entire enterprise's information systems.

Kerberos requires a user to take three steps to access a data, each involving a message that needs to be exchanged with the database. This message is called a ticket. The three step ticket exchange strategy in Kerberos may be explained as-

* Client Authentication- The Authentication server authenticates the user by providing a time stamped Ticket-Granting- Ticket (TGT).
* Authorization- The user uses the TGT to request a service ticket from the Ticket Granting Server.
* Service Request-The database providing the service to the user authenticates the user using the service ticket.

The key distribution centre in Kerberos comprises an Authentication server and the Ticket Granting Server.

**Implementation of Biometric templates in Kerberos Authentication Protocol**

Biometric in the Authentication will add more security to the data. Biometrics is the technique which utilizes the individual's physiological characteristics like the iris, fingerprint, pupil, etc or the behavioral characteristics such as the signatures, gait, etc to achieve identification or recognition. The biometrics basically prevents stealing of possessions that mark the authorized person's identity i.e. security badges, licenses or properties. Every biometric prototype should essentially be standardized, special, permanent and appropriate to the user, providing collectivity, consistency and resistance. It contains two-layer watermarking to ensure that the hidden channel is safe. This approach promotes high protection and privacy, ensuring reversibility, precision and diversity by virtue of the recoverability of multi-line code (MLC). (Radhika, Apoorva & Nidhi, 2015)

Initially there was no security in distributed databases. It didn't authenticate neither the users nor the services and there was no data privacy. Using Kerberos through biometric templates provides the following ACI properties to be established in the security architecture of distributed database-

* Availability- Due to the efficient scalability features provided by distributed databases the information is always available to the intended users.
* Confidentiality- only the intended users may access the services by the server. High secure services are offered through the biometric templates that are not easy to impersonate providing an added advantage over the alphanumeric character passwords which could be easily hacked or disclosed.
* Integrity- The files exchanged over the system in this encrypted format ensures that there is no alteration of the messages exchanged over the network providing the authenticated message exchange over the network.

**2.2 Access Control**

Access control guarantees that only authorized users perform operations they are allowed to perform on the database. Many different users may have access to a large collection of data under the control of a distributed database. The distributed DBMS must thus be able to restrict the access of a subset of the users. Access control in the database differs in several aspects from that in traditional file systems. In the access control approach, authorization must be redefined so that different users have different rights on the same database objects. This necessity means that subsets of objects can be specified more precisely than by name and between user groups. Furthermore, in a distributed setting, the decentralized control of permissions is especially important.

In the access control approach, the organizations must need to consider the Access Control Policy. Access control requires the enforcement of persistent policies in a dynamic world without traditional borders. Most work in a hybrid environment where data moves from on-premises distributed servers or cloud to offices, homes, hotels, cars and coffee shops with open wi-fi hotspots, which can make enforcing access control difficult. In the past, access control methodologies were often static. A sophisticated access control policy can be dynamically to respond to evolving risk factors, enabling a company that’s been breached to “isolate the relevant employees and data resources to minimize the damage”.

The primary concern of an access control system are the following:

1. Prevent access: in the absence of any privilege, ensure that the subject cannot access the object.
2. Determine access: Decide whether a subject has access, according to some policy, to take an action with an object.
3. Grant access: give a subject access to an object.
4. Revoke access: remove a subject’s access to an object.
5. Audit access: Determine which subjects can access an object, or which objects a subject can access.

**2.2.1 Discretionary Access Control**

Discretionary access control or authorization control defines access rights based on the users, the type of access and the objects to be accessed.

DAC policies include the file permission model implemented by nearly all operating systems. The authorization can be viewed as a triple (subject, operation type, object definition) which specifies that the subject has the right to perform an operation of operation type on an object. To control authorization properly, the DBMS requires the definition of subjects, objects, and access rights. Authentication credentials such as username and password are verified before access is granted. This type of access control is highly flexible in terms of data control. It gives room to customize access policies according to each end-user. Access is read and written to each user a single file. It creates a firewall against malware attacks, unauthorized access by up to a highly encrypted security protocol that must be bypassed before access is granted. This goes further to increase reliability in the organization. (Ahn, 2009)

**Features of Discretionary Access Control**

Flexibility**:** It allows users to customize their access policies individually.

Ease of Control: The security system allows easy monitoring of the access points. This is done using DAC devices such as keycards to permit and monitor particular positions of the organization.

Usability:It allows easy policing and granting permissions for each access point. The complexity of access control is minimized to achieve better management of the network’s resources.

**Limitation**

* Conflict with relational dependencies
* Increased complexity in order to preserve both relational dependencies and security dependencies

**2.2.3 Multi-level Access Control**

Multi-level or mandatory access control further increases security by restricting access to classified data to cleared users. Also it is the solution for limitation of Discretionary Access Control. It stems from increased security threats coming from the internet. It is a policy in which access rights are assigned based on regulation from a central authority. Multilevel security prevents unauthorized users from accessing information at a higher classification than their authorization. It also prevents users from declassifying information.

Working principle is the Bell-LaPadula Model in the multilevel access control. The model focuses on data confidentiality and controlled access to classified information. In this formal model, the entities in an information system are divided into subjects and objects. The Bell-LaPadula is built on the concept of a state machine with a set of allowable “secure states”. The notion of a “secure state” is defined, and it is proven that each state transition preserves security by moving from secure state to secure state. This inductively proves that the system satisfies the security objectives of the model. A process has a security level also called clearance derived from that of the user. In its simplest form, the security levels are Top Secret (T S), Secret (S), Confidential (C) and Unclassified (U), and ordered as T S > S > C > U, where “>” means “more secure”.

Access in read and write modes by subjects is restricted by two simple rules:

**Rule 1 (called “no read up”):** A subject S is allowed to read an object of security level l only if level(S) ≥ l. That means a subject at a given security level can only read objects at the same or lower security levels. For instance, a subject with secret clearance cannot read top-secret data.

**Rule 2 (called “no write down”):** A subject S is allowed to write an object of security level l only if class(S) ≤ l. That means a subject at a given security level can only write objects at the same or higher security levels. For instance, a subject with top-secret clearance can only write top-secret data but cannot write secret data (which could then contain top-secret data).

Multilevel access control has a strong impact on the data model because users do not see the same data and have to deal with unexpected side-effects. One major side-effect is called polyinstantiation which allows the same object to have different attribute values depending on the users’ security level. Polyinstantiation can occur in two possible ways. They are called visible and invisible polyinstantiation.

Visible polyinstantiation occurs when a subject with higher level of clearance tries to insert information in a field that already has that information, but with lower level of sensitivity. By changing that value, one of basic principles of the mandatory model of access control would be violated.

Invisible polyinstantiation occurs in a reversed situation, when a subject with lower level of classification tries to insert information in a field that already has that information, but with higher classification. If that change was to be declined, the system would confirm the existence of information with higher levels of classification. (Andro & Vlatka, 2007)

Can implement multilevel security with the following combinations:

**Database authorization with multilevel security with row-level granularity**

In this combination, the database administrator gives authorization at the object level. Multilevel security is implemented only at the row level within the database.

**External access control and multilevel security with row-level granularity**

In this combination, external access control (such as the RACF® access control module) is used for authorization at the database object level. External access control also uses security labels to perform mandatory access checking on database objects as part of multilevel security. Multilevel security is also implemented on the row level within the distributed database.

**Limitation**

* For every query, the security level of every element needs to be checked with the security level of the user. Hence, the process is a bit slow.
* Care has to be taken for some special conditions which might arise during “write up” operations.

**3. Comparison of Issues and Approaches**

Here we come up with the different approaches we have studied and the pros and cons and the difference between the approaches.

**3.1 Key difference of reviewed Approaches**

Here the first property a user with low access is not allowed to view an object with high access. Using multilevel access control in such cases restricts our self to add flexibility. Also a super user has the legitimate rights to perform write operation on low access objects. But care has to be taken to avoid leakage of information. But unlike MAC, Discretionary Access Control is more flexible. DAC is based on granting and revoking of privileges. Privileges here are assigned at two levels. Each user account is assigned rights or privileges and the relations are also assigned privileges is - the privilege to access a particular is controlled or restricted.

Compared with the authentication and the access control, authentication is used to verify the identity of a user who wishes to access the system. Because access control is typically based on the identity of the user who requests access to a resource. So authentication is essential to effective security. Without authentication, access control can not be possible. So to secure the data, authentication protocol has to be impactful and fully essential to support the database integrity and availability.

**3.2 Key difference of security issues**

Replication is considered to overcome database failure. Data has been stored at more than one site or node. It is useful in improving the availability of data, in case of failurance of the site or node. Speed up the query evaluation. Distributed databases should be able to survive in a system failure without losing any data. However Fault tolerance is made possible by the partitioned architecture of the system and data redundancy therein. Fault-tolerant algorithms or systems use backup components that automatically take the place of failed components, ensuring no loss of the transaction or operation. Where concurrency control permits users to access a database in a multi-programmed fashion while preserving the illusion that each user is executing alone on a dedicated system. It ensures that multiple transactions are executed simultaneously while maintaining the ACID properties of the transactions and serializability in the schedule.

**4. Security Challenges in Distributed Systems (Unsolved Problem)**

In different fields, such as healthcare and homeland security, all organizations, from commercial to social institutions, may incur substantial financial and human losses as a consequence of unauthorized data observations.

**4.1 Appropriate Access Control:**  Take data storage systems into consideration. The key concerns are the application of effective access management procedures in the processing of data from the warehouse and the safekeeping of a data storage network from the downstream information systems. For eg, security policies for the different data sources that must be integrated from the warehouse to form a warehouse policy. During the transition, the security strategy of the warehouse needs to be followed and the warehouse audited. The question of storage then is a matter for the warehouse, for example, who will save average wages to enter the warehouse and then subtract the different wages from confidential data base. The question of deduction could thus become a warehouse concern. To date little work on security and the issue of recovery for the warehouse has been published. This area requires a great deal of study. (Radhakrishnan, Ganesh, Akila. (2017)).

**4.2 Security Problems with Distributed Data Mining:** Data mining poses extreme security issues, e.g. whether a machine is able to use data mining software. The user should take on different questions and establish a delicate theory that helps for data recovery. This problem can be dealt with in different ways. Through a database and a specialized data mining method, users may submit the method and a question can be extracted if confidential details can be obtained from unclassified information. One of the issues with this method is that consumers can only use that tool, but still have many available methods at their hand. In fact, the recovery problem can not be addressed in any way. Another approach to the recovery problem is to create a rest and recovery system to detect user patterns and prevent the problem. This controller lies between the data extraction tool and the DBMS-managed database or data source. In the distributed world, the data mining method is being expanded to function. This program is called the distributed data mining method and is focused very little.(Radhakrishnan, Ganesh, Akila. (2017)).

**4.3 Concurrency Control Performance:** Competition management strategies have historically been divided into four groups – locking, timestamp ordering, positive and combination. Within this paper we will look at some of the proposals and related problems of each division. Currency management methods have been analytically explored in distributed libraries, although these computational experiments have been limited. The latest Two-Phase Locking (2PL) computational models in a distributed database do not require simultaneous transaction processing at multiple locations. The basic timing strategies for ordering were also studied. It was recorded that BTO performed in a distributed database worse than 2PL, especially in a low data containment scenario. Deadlocks are prevented through the use of timestamps in the case of wound-wait (WW). Due to the original start-up date, each transaction is numbered; younger transactions can not make older transactions wait. When an older transaction requires a lock and the request results in the older transaction waiting for a younger transaction, the younger transaction is wounded, unless the second stage of its commit protocol already contains a restart. A major downside in WW avoidance techniques is that a subpoena for a data object found in another transaction is not a deadlock. The related expenses for the response are also incurred. There may also be no need for abortions and rollback. For Distributed Certificates (OPT), restarts are often used to resolve conflicts, and then search for issues when a committed transaction is ready. This means it slows cohort-update contact before commit time. It can be lethal to back the strategic tracking details from the commit message.(Akintola, AA, Osakwe. (2005)).

**4.4 Common Network Attacks in Distributed Systems:** In the distributed environment, the following are typical forms of network attacks. Passive Attack, An Active Attack, Syn Flood Attack, Password Attack, Distributed Attack, An Insider Attack and Phishing Attack

A passive attack typically monitors unencrypted messages, finds simple passwords and personal information which can be used in other attack styles. Passive attacks include traffic analysis, unprotected communications monitoring, weakly encrypted decryption of traffic, and authentication data collection such as passwords.

The attacker attempts to circumvent or break into secure systems in an active attack. It can be done by deception, malware, worms or Trojan horses. Active attacks included attempts to bypass or break protective functions, malicious code introduction and information stealing or alteration.

A distributed denial of service ( DDoS) effort to use the power of the target to keep it from delivering support. The amount of resources used are one way to identify DDoS attacks. In general, the resource used is an internal target system host resource or a data capacity transmission in the network to which the target is assaulted. The SYN flood attack is a clear example of a resource attack.

An in-company assault includes people from inside, including an unlucky person who may target the network Intruder hacks, whether they are malicious or not. Intentionally malicious insiders eavesdrop information, steal information or harm it; use information fraudulently; or refuse access to other authorized user information.

When phished, the hacker produces a fake website that resembles a famous website business like Facebook, Hotmail, Yahoo or consumer goods businesses. The attack is done by the hacker who then sends an e-mail to the user to click a link to the wrong website. (Shadmanov, Istam & Shadmanova, Kamola. (2016)).

**5. Methods To improve**

Multi-level access protection, confidentiality, stability, integrity and recovery are some of the most critical safety standards for database management systems. Therefore, the full data security approach requires to satisfy the following three requirements: 1) the security of data from unwanted access, the privacy or confidentiality of data, 2) the integrity of unwanted and unsuccessful data modification, and 3) the availability of data. (Ozsu,Valduriez.(2011)).

**5.1 Hybrid Lock Buffer:** This hybrid technique is determined by the design of the lock manager. A final lock buffer is retained by the lock holder. The lock buffer includes keys and pending demands for locking of a single data object for any slot or system. Therefore, the number of active lock request data objects can not exceed the number of buffer spaces. If the lock manager gets a lock request for an element x, the lock manager tries to find x in a lock buffer first. If the lock manager is identified, he tries to place the lock in the door. Depending on current locks status on x, the request for locks can be granted or denied in the same way as pure locks.

If the capacity of the lock buffer is zero, all lock requests will be rejected and no active locks will ever occur. All transactions in this scenario are optimistic about all data items in your read and write sets and the system is purely optimistic. On the other hand, whether the number of slots is greater or equal to the number of data items in the database, and lock query may be rendered without any current data items and locks being deleted. The transaction must be checked when committed. To be legitimate, a transaction must be legitimate in its read and write sets with respect to all data objects.

The contract shall be checked until pledged. In order to make a transaction legitimate, all data elements in its read and write sets must be legitimate. Transactions T are true for an element X of a data if one of the three following conditions is present (a transaction T To T which is executed overlaps A; that is, T' occurred after T, but prior to T, or T occurred after T, and before T' began its validation phase) (T' is true for a transaction T').

As the empirical findings show, a relatively small lock buffer is used to improve the performance of optimistic competitive controlling techniques significantly. The hybrid system thus achieves our aims of enhancing ambitious competitors' efficiency and, in many ways, is even better than 2PL. (Akintola, AA, Osakwe. (2005)).

**5.2 Using Convert Channel:** Multi-level access protection, confidentiality, stability, integrity and recovery are some of the most critical safety standards for database management systems. Therefore, the full data security approach requires to satisfy the following three requirements: 1) the safety of data from unwanted access, the privacy or confidentiality of data, 2) the quality of unwanted and unsuccessful data modification, and 3) the availability of data.

A DBMS that complies with these mandatory requirements is able to offer safety at various levels. But multilevel security, mainly on the distributed environment, becomes a major issue to allow replicated data to be accessed at various locations by several users simultaneously. Stable multi-level database systems have a number of specifications above traditional database systems. There are several conceptual models that specify transaction access rules for secure database systems. The Bell La Padula model has a major role to play. In this model, transactions and data are given a protection standard. A transaction protection level reflects the level of clearance and for records, the level of security is the level of classification. Transactions can not be read at a higher level of security or write at a lower level of security. Thus, high security data will implicitly be moved to the lower level of security by avoiding low security transactions by necessity. This is referred to as a covert channel. Any components or functions in a network misused for encoding or representing information for illegal transmission are covert channels without infringing upon the specified access control policies. Covert channels are paths that are not usually used in the flow of data. A low security transaction can be postponed or aborted in multilevel secure repositories by mutual data access by a high security level transaction. (Batra, Singh. (2011)).

**5.3 Blockchain Technology:** Blockchain is a "immutable decentralized leader of transactions held by a centralized consensus protocol over a peer-to - peer network".

Blockchain's implementation in the banking industry is likely to overcome logistical shortcomings and reduce overhead and transaction costs. This protocol eliminates the risk of human error and manipulation by requiring intelligent contracts to do much of this work without human intervention. Blockchain is able to minimize or even remove the need for third parties to guarantee the integrity of financial transactions. Legal arrangements can be accelerated as money transfers are no longer required for days. Blockchain will be able to monitor land records and check ownership documents with transaction timestamps. Blockchain could be used in education to verify and confirm student attendance, averages grades and degrees. In order to validate candidate certificates, this could save time and resources. The public layer would include information required to show user honesty and credibility, which would include the qualifications, courses which grades the student in the private layer. This protocol ensures that these data are confidential and complete.

Blockchain plays a crucial role in the management of medical records and patients data in healthcare. The healthcare data platform allows patients to access and monitor the healthcare records they want to share with each provider. The opportunity to exchange clinical data with various agencies allows "a greater knowledge of public health and illness rates and trends, to ensure improved patient treatment, improve preparation or physicians advice, and prepare programs that best benefit from small domestic health sector budgets, the well-being, etc. (Zahadat, Partridge. (2018)).

**5.4 Wound Wait:** The 2PL protocol is implemented by the same procedure. This algorithm treats the deadlock problem differently. The timestamps in this algorithm are not used to "wait for" details, as in the 2PL, for checks on local and global blockades. Each transaction is counted according to the original start-up date. The younger transactions are not allowed to wait for the older transaction. If an older transaction requests the lock, and the older transaction requires that the lock await a smaller transaction, the younger one is injured. It is restarted until the commit protocol is in the second step. It is restarted so that the commit protocol will not be in the second step. If the younger transaction is waiting for an older transaction, the risk of deadlocks will be removed. (Sinde, Aware. 2016).

1. **Summary**

This section summarizes the studies about the Security of distributed systems in databases and the methods to overcome that as reviewed from different papers.This report discusses the most popular emerging security mechanism and issues in distributed database networks. When a distributed database becomes more popular, there is a much greater need for improvement in distributed database management systems. (Akintola, Aderounmu, Osakwe (2005)). The most critical problem is security that might emerge and potentially jeopardize access control and network integrity. In this paper, we are proposing some solutions for certain security aspects, such as multi-level access control, confidentiality, reliability , integrity and recovery related to a distributed database system. We reviewed around 20 scholars' papers to understand the issues related to security in distributed database systems and proposed solutions for the current ongoing security issues.

We have learned and used certain problems and the approaches according to the situation, the distributed databases were facing and the approaches and algorithms derived to solve, also covered the recent issues facing the multinational companies like Google, Facebook, Amazon and so on for using the distributed database. Searched the companies websites to learn their problems and approaches they have developed or working on it.

1. **Is Review Comprehensive?**

We learn about different mechanisms such as access control, authentication control, concurrency control techniques (Oszu, Valduriez. (2011)). The methodological approach and how this method helps in the security of the data in the distributed systems. The study also describes the unsolved problems and challenges that are currently faced in the distributed database security environment. It is not easy to achieve the property in distributed file systems.

**7.1 Conclusion**

There are many approaches that are proposed but the problems with the security of the distributed systems is always a question. Day by day the challenges and the attacks arise.

Based on the study of the different papers we concluded that the study conducted by us from the different papers were very comprehensive. We learn about different mechanisms such as access control, authentication control, concurrency control techniques (Oszu, Valduriez. (2011)). The methodological approach and how this method helps in the security of the data in the distributed systems. The study also describes the unsolved problems and challenges that are currently faced in the distributed database security environment. It is not easy to achieve the property in distributed file systems. Many factors are important and typically these factors are not predictable. The paper also describes the common problems that are faced and the problems with the authentication and access control methods.

The suggested course of action and improvements to be done in the security of the distributed environment is also proposed in this paper so that the system becomes more strengthened and adds another layer of security. The duplication of files (backup node) and avoiding potential errors. How using the covert channel a low security transaction can be postponed or aborted in multilevel secure repositories by mutual data access by a high security level transaction. (Batra, Singh. (2011)). Also the study proposed introducing methods of adapting Blockchain Technology that not only decrease the gaps of the vulnerabilities but also provides a more reliable and trusted environment. The paper also describes how implementing a wait and wound method can help to avoid the problem of the deadlocking during the transaction. (Sinde, Aware. 2016).

**7.2 The Scope**

This review will look at the various aspects and ways of dealing with the fundamental security issue in the distributed database systems. This paper starts by reviewing the simple distributed database systems available commercially and their embedded features and continues on with popular structures built on them to guarantee or enhance the security of the distributed database system underlying them.

Like the distributed file systems, the security area is quite diverse and therefore this paper will focus on a number of key examples highlighting the wide array and types of solutions to solve database system security problems.

Authentication, access control, concurrency control, blockchain,trust-based models etc. are many developments towards the generation of secure and trusted distributed systems. The writers must establish modern and successful strategies in future to design stable distributed systems .

**References**

Köse, İ. (2002). Distributed Database Security. Data and Network Security-Spring. Retrieved from <https://pdfs.semanticscholar.org/049b/f9271f5b01e0d1b9fd0379cc9846fd31a21e.pdf>

Radhakrishnan, Sangeetha & Ganesh, Akila. (2017). Security Issues in Distributed Database System. 4.

Akintola, AA & Aderounmu, Ganiyu & Osakwe, AU & Adigun, Matthew. (2005). Performance modeling of an enhanced optimistic locking architecture for concurrency control in a distributed database system. Journal of Research and Practice in Information Technology. 37.

Ozsu,Valduriez.(2011) Distributed Data Management:Unsolved Problems And New Issues Retrieved from

<https://pdfs.semanticscholar.org/4b1a/aea313c50bcaba42cf8e5a8eef83aae5dab1.pdf>

Batra N., Singh M. (2011) Multilevel Policy Based Security in Distributed Database. In: Abraham A., Lloret Mauri J., Buford J.F., Suzuki J., Thampi S.M. (eds) Advances in Computing and Communications. ACC 2011. Communications in Computer and Information Science, vol 190. Springer, Berlin, Heidelberg

N.Zahadat, W.Partridge. (2018) Blockchain: A Critical Component to Ensuring Data Security. Department of forensic sciences, University of Baltimore, USA, vol 10.

Retrieved from <https://juniperpublishers.com/jfsci/pdf/JFSCI.MS.ID.555780.pdf>

Quasim, Mohammad. (2014). Security Issues in Distributed Database System Model. Indian Stream Research Journal. 2. Retrieved from <https://www.researchgate.net/publication/315459325_Security_Issues_in_Distributed_Database_System_Model>

Yunus, M. A. M., Krishnan, S. K. G., Nawi, N. M., & Surin, E. S. M. (2017). Study on Database Management System Security Issues. JOIV : International Journal on Informatics Visualization, Retrieved From <https://www.techrepublic.com/resource-library/whitepapers/security-issues-in-distributed-database-system-model/>

Z. S. Zubi, "On distributed database security aspects," 2009 International Conference on Multimedia Computing and Systems, Ouarzazate, 2009, pp. 231-235.

Dr. A.Akila.(2016). Security Issues in Distributed Systems, International Journal of Engineering Research in Computer Science and Engineering. Retrieved from

<https://www.researchgate.net/publication/325737373_Security_Issues_in_Distributed_Database_System>

D.S.Hiremath, Kishor (2016). Distributed Database Problem areas and Approaches, IOSR Journal of Computer Engineering (IOSR-JCE) <https://www.iosrjournals.org/iosr-jce/papers/conf.15013/Volume%202/4.%2015-18.pdf?id=7557>

Hamed Almutairi, Abdulrahman Helal Alruwaili(2012). Security in Database Systems, Global Journal of Computer Science and TechnologyNetwork, Web & Security. Retrieved from <https://globaljournals.org/GJCST_Volume12/3-Security-in-Database-Systems.pdf>

A. Galinovic and V. Antoncic, "Polyinstantiation in Relational Databases with Multilevel Security," 2007 29th International Conference on Information Technology Interfaces, Cavtat, 2007, pp. 127-132, doi: 10.1109/ITI.2007.4283757.

A.A.Akintola,G.A.Aderounmu and A.U.Osakwe,” Performing Modeling of an Enhanced Optimistic Locking Architecture for Concurrency Control in a Distributed Database System”,ACM vol.37, No.4, November 2005

R. Maheshwari, A. Gupta and N. Chandra, "Secure authentication using biometric templates in Kerberos," 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, 2015, pp. 1247-1250.

Mohammad A., Abdelfatah Aref, and Omaima N. A. AL-Allaf, “Cryptography Based Authentication Methods,” 2014 Proceedings of the World Congress on Engineering and Computer Science 2014 Vol I, San Francisco, USA. Retrieved from

<http://www.iaeng.org/publication/WCECS2014/WCECS2014_pp199-204.pdf>

Shadmanov, Istam & Shadmanova, Kamola. (2016). Summarization of Various Security Aspects and Attacks in Distributed Systems: A Review. Advances in Computer Science. 5. 35-39.

Vinayak Sinde, Preeti Aware. (2016). Concurrency Control In Distributed Database Systems: International Journal for Research in Engineering Application & Management (IJREAM). Vol-01. Retrieved from <https://www.ijream.org/papers/INJRV01I10005.pdf>

Ahn GJ. (2009) Discretionary Access Control. In: LIU L., ÖZSU M.T. (eds) Encyclopedia of Database Systems. Springer, Boston, MA

M.T. Özsu and P. Valduriez, Principles of Distributed Database Systems: Third Edition, DOI 10.1007/978-1-4419-8834-8\_5

<https://link-springer-com.proxy-bc.researchport.umd.edu/content/pdf/10.1007%2F978-1-4419-8834-8_5.pdf>