# Security and Privacy Solutions associated with NoSQL Data Stores

## Introduction

The article emphasizes the need of NoSQL databases as a result of developments in Distributed web applications and Cloud computing. Technological organizations like Amazon and Google created their own NoSQL, distributed storage system like Dynamo Technology and Bigtable to cope up with the ever-increasing large volume of data for storage and further processing. Secondly, modern companies heavily on rely on database system which is not relational and can encounter scalability and availability problems which classical database system like SQL is not able to fulfill. Cap theorem is introduced which suggests that no shared data system can provide Consistency, Availability, Partition Tolerance all at the same time. The gist of this paper is to focus on the security and privacy issues with NoSQL databases and examine it to deliver solutions to tackle the issues.

## Related Work

This section deals with the multiple research that has been done in the domain of establishing relationship between Relational and NoSQL databases. Some of the early papers bolsters the usage of NoSQL for the large, public and content centric applications whereas also emphasizes that relational database will not go anytime soon and can be implemented for the business operations. The usage and benefits of using NoSQL databases like Cassandra has been discussed, like how they scaled the scaled the network without compromising hardware and server infrastructure. Some papers focused on enhancing the privacy in NoSQL databases and provided solution for the same. Organizations took help of proxy and standard encryption method for data protection and encoding techniques for indexes.

## Comparison of Relational and NoSQL databases

This part contrasts relational databases like SQL with NoSQL databases, with a focus on the benefits and distinctions of each type of databases. NoSQL databases has large spectra of benefits to offer like schema less data representation, speed, elasticity of the applications and reduction of development times as developers does not have to deal with complex SQL queries.

NoSQL databases unlike relational databases do not fully support the ACID model and may sacrifice some of these properties for scalability or performance. But NoSQL has its own merits of not following ACID model due to which it is fully compatible with cloud technology as they are able to analyze any type of data. Relational databases do horizontal scaling for upgrading the performance while it is vice-versa for non-relational databases. NoSQL databases are less complex than relational databases because there are no implications to create tables to record data which is a bit complicated task. Non-relational databases have a huge role in managing Big Data due to their ability to store and retrieve data quickly across distributed nodes, thus making use of multi-core GPU architectures. As NoSQL is schema less and table less it can easily handle other forms of structured or unstructured like word, pdf, images and many more. Relational databases use recovery manager for ensuring durability and transaction atomicity by using log files and ARIES algorithm. However crash recovery depends on replication to recover from the crash. Both relational and non-relational databases don't provide embedded security in the database itself. However, cryptography mechanisms can be implemented in the security systems middleware for the relational database. On the other hand, there are solutions which provide security mechanisms, but it compromises scalability and transparency of the NoSQL system. That's why only few NoSQL databases provide in-built encryption environments.

## Proposed Security and Privacy Solutions

Pseudonyms-based Communication network are very effective in enhancing security for the NoSQL data stores. Users need to enter their credentials for only one time and one can access multiple services by using pseudonyms provided to them. As a result, it supports the integrity of the transactions by disclosing their identity. The system is based on the RSA and Diffie Hellman cryptography protocols extended by Brand's Credential System along with four parties: the Users, a central identity provider, service providers and organization for issuing and validating credentials. Service providers know the users by their pseudonyms, hence maintain privacy. Each credential is encoded with unique public and secret key under the Discrete Logarithm assumption. Corresponding user can prove the ownership of the digital pseudonym to another organization without revealing secret key or personal information. The system offers unlikability, which prevents the linking of two pseudonyms belonging to the same user, instead it identifies by the set of credentials. Although the service providers associate user with a different encoded random number of their pseudonym each time, they also have the authority to blacklist the user if a user makes abuse of the service. However, these available applications cannot detect and disable malicious jobs and queries, and the Kerberos central authentication system can be easily surpassed with the advanced scripts. Real-time security mechanisms exist in big data technology but are limited to controlling user requests at the API level. But one suggested technique to tackle this problem would be to use Kerberos for the initial authentication followed by second level authentication for accessing MapReduce.

## Conclusion

The paper focussed on security concerns of NoSQL database along with the comparative analysis with the relational databases. Reasons for the security issues like Data protection and access control, along with security threats have been thoroughly discussed. NoSQL databases use Kerberos to authenticate the clients and nodes whereas Cassandra uses Transparent Data Encryption (TDE) to encrypt the data before it is written to disk and decrypted when it is read from disk. Administrator must implement access control mechanisms for the data access among users to maintain a secure MongoDB deployment. Also, some techniques like Pseudonyms based communication network and monitoring techniques have been discussed to mitigate the attacks on NoSQL databases.

## Scope of Improvements

- Solutions to mitigate the attacks on NoSQL databases are provided without any practical implementation.
- Lack of evidence for the facts. For example: In the statement, "providing security mechanism to NoSQL database would compromise scalability and transparency of the system", author states that without mentioning how.
- They could have written something about how the involvement of emerging techniques like machine learning and blockchain could shape the privacy of NoSQL.

## References

1) "How to use Cassandra with TDE Transparent Data Encryption," *CopyProgramming*. [Online]. Available: https://copyprogramming.com/howto/how-to-use-cassandra-with-tde-transparent-data-encryption. [Accessed: 02-Apr-2023].