A Secure Cloud Service for Managing User's Crucial Data Using NLP, Blockchain, and Smart Contracts

Arun Kumar Banavara Ramaswamy * , Komala Rangappa , Mahadeshwara Prasad , Shreyas Arun Kumar

Posted Date: 23 September 2024

doi: 10.20944/preprints202409.1738.v1

Keywords: Cloud computing; Big data; Sharding concept; Blockchain; Smart contract; Crypto wallets; Natural Language Processing; Crucial Data privacy

Article

A Secure Cloud Service for Managing User's Crucial Data Using NLP, Blockchain, and Smart Contracts

Komala Rangappa [1,2,†], Arun Kumar Banavara Ramaswamy [1,3,*,†], Mahadeshwara Prasad [3,†] and Shreyas Arun Kumar [4,†]

[1] VTU Research Center, Department of Master of Computer Applications (MCA), BMS Institute of Technology & Management, Bengaluru 560064, India; komal.uday@gmail.com
[2] Department of Computer Applications, M.S. Ramaiah Institute of Technology, Bengaluru 560054, India
[3] Department of Computer Science & Engineering, BMS Institute of Technology & Management, Bengaluru 560064, India; mahadeshwara.prasad07@gmail.com
[4] Department of Computer Science & Engineering, Sai Vidya Institute of Technology, Bengaluru 560064, India; shreyasvasista05@gmail.com
* Correspondence: arunkumarbr@bmsit.in; Tel.: +91-98-8600-8210
† These authors contributed equally to this work.

Abstract: The management of user data in the cloud is easily poised to become a giant issue for any business, with so much digital information floating around these days that open it up as an easy target when companies aren't vigilant. In this paper, a novel cloud-based service is proposed that employs various advanced NLP and encryption methods with the use of blockchain. These techniques are combined to provide a solid solution to secure fundamental data such as credit card numbers, passports or any government identity cards etc. Using a hybrid NLP model, integrating Transformer Models and Named Entity Recognition (NER), to automatically catagorzing data as critical vs non-critical. Only the most important data is encrypted by a user's cryptographic wallet before being divided into multiple chunks and stored on an exclusive cloud cluster; metadata then takes turns managing securely through blockchain to provide traceable means of retaining integrity. Smart contracts provide strict access control measures and change the cryptographic nonce if need be to prevent illegal entrance into a specific zone, thus create security. This all-inclusive strategy maintains well-known high security standards for protecting the confidentiality, availability and integrity of your sensitive data on a global scale—delivering you simple yet scalable secure world-class cloud-based data management. A proposed framework developed to fulfil the security requirements for current cloud services, which is a beneficial contribution in context of data protection and cloud Security.

Keywords: cloud computing; big data; sharding concept; blockchain; smart contract; crypto wallets; natural language processing; crucial data privacy

1. Introduction

In the cyber era of threats, it is a necessity to keep crucial information well-managed and protected. Data like credit card numbers, passport details or government-issued identity cards holds utmost importance today! It leaves you dangerously dependant on cloud services, and raises the stakes of storing and handling sensitive data online. Today, cloud systems provides many ways of data storage however they do not provide a reliable way to manage our crucial information securely. Indeed, traditional data rest and recovery techniques are prone to advanced threats such as cyber attacks, second-party security breaches, non-authorised access which may cause potential financial loss or breach of privacy. As more cloud-based services proliferate, this is becoming an increasingly important area of challenge and a problem requiring innovative solutions to maintain the privacy, integrity, and availability for sensitive user data while benefiting from the scalability and efficiencies available in these environments.

In order to tackle these issues, it makes effective use of sophisticated Natural Language Processing (NLP) techniques that classify the user data properly. The system uses Transformer Models in combination with Named Entity Recognition (NER) for automatic classification of data into critical

 doi:10.20944/preprints202409.1738.v

and non-critical citing units. This provides much finer-grained classification off data (reducing the probability of wrongfully handling sensitive info). NLP models have a few benefits, e.g., efficient way to read and understand great amounts of text data with no format structure (unstructured), more natural in every context making learn through ML for an ongoing evaluation. All these functionalities put NLP as an essential resource to increase data security at cloud environments, being vital in the correct identification and treatment of important information according with their importance.

After classifying the data, crucial data is encrypted by a cryptographic wallet of user. This encrypted data is then hashed to produce a unique reference key which maps to the non-crucial data. For added security, this hash will changes dynamically every time the user accesses their data. This change can be done in two ways:choose a random value nonce or increase the nonce by 1 with each access attempt. By constantly varying the hash like this, it becomes nigh impossible for a malicious actor(s) to ascertain or even predict where crucial information can be found. The higher the nonce harder the complexity of deciphering the hash also grows, providing an additional layer of protection against unauthorized access. As the nonce increases the hash difficulty also increases, hence validation and retrieval of crucial data requires more computational energy. In order to solve this, the nonce will be converted back to zero once the system detects the difficulty of the hash has increased to a certain level where it needs a lot of computation energy.

Blockchain-based smart contracts (SC) are also used to make the data management process more secure. SC automatically executes a specific clause of contract when the conditions defined are met. It serves as on the fly auditing , enhancing security and providing tamper proof access controls. The blockchain verifies all actions and records as transactions on a decentralized ledger. When a user wants to access the data, for example, the smart contract verifies if that request should be allowed or not and monitors any suspicious access. If unauthorized activity is detected, the IP address and location are recorded along with a notification to both user and cloud service provider. Smart contracts enhance security with access control and also provide transparency because all actions are visible on the blockchain ledger, which is immutable.

Our system is a complete solution for handling key data in the cloud, it combines NLP to classify data with encryption and hashing of added value solutions as well smart contracts controlling access. By combining all relevant modules, this approach helps in proper classification of sensitive data securely managing it under tight control thus establishing a comprehensive foundation that solves the issues inherent to current systems. It improves data privacy and reduces security risks, all while providing a platform for growing organizations to defend against new attack vectors — becoming the gold standard of cloud-based data management.

The significant contributions in this paper:

- This paper introduces a new framework based on transformer model and NER to improve the accuracy of data classification (crucial vs. non-crucial) concerning user security in cloud environments, which borders irregularities associated with managing registered databases efficiently.
- It offers a new way to secure critical data with encryption addressed by the user and also serves as hash-based in nature, which change every time one access thus creating two nonces for nonce selection that aide no unauthorized tracking or accesses from malicious users.
- The proposed system integrates blockchain-based smart contracts for automated, tamper-proof access control with transparency and its associated reliability guarantees a robust defense against intra as well inter breach actors.
- The holistic nature and scalability of the proposed solution to handling sensitive user data in current state-of-the-art cloud environments distinguish it from a limited set of complimentary solutions but as part of an overall security strategy involving NLP, encryption, hashing end-to-end with blockchain smart contract

 doi:10.20944/preprints202409.1738.v

2. Related Work

Modern evolutions of cloud storage and data governance continue to focus on more clever ways to secure private information for sensitive areas. Integration of Natural Language Processing (NLP) models like Named Entity Recognition and Transformer-based architecture have been researched to identify and categorize specific data residing within unstructured text, eventually aimed at improved precision as well as context understanding. On the other hand, cryptographic solutions, [1–25] distributing key parts of sensitive information across multiple servers to make sure no single server can be compromised for its private keys or hold all gathered datasets. To obscure attack vector points, dynamic nonce-based hashing schemes have been made introducing to keep changing hash of data in every access that contribute higher integrity and confidentiality. Smart contracts and blockchain technology helping with transparent mechanisms of access control that are immutable as well compliance for regulatory standards while decreasing reliance on central entities.

Cloud Storage Security and Data Management Techniques have advanced a lot over years but it has several critical gaps. Current methods that use Named Entity Recognition (NER) models do not have sufficient contextual information to distinguish between essential and non-essential data, most notably when dealing with complex or ambiguous factors. While more recent work has proposed Transformers to achieve this contextual understanding, in the domain of cloud classification data set and as far I know there is limited research combining NER with Transformer models. In addition, although some form of hashing and encryption has existed to protect confidential information from being read by unauthorized personnel for years, the old-school (pen-and-paper) ways are often unable or fail dynamically adapt against access patterns that can leave doors open to data siphoning strategically carried out with finesse over time.

And, in most cases this integration of blockchain technology and smart contracts has only been superficial as there is no real-time tamper-proof secure access control or auditing mechanism. However, the integration of smart contracts in hybrid systems that combine cryptographic approaches with NLP has not been explored up to our best knowledge for automating access control in a secure manner and reducing dependency on central authorities. This is what our project aims to achieve a new unified framework that combines the power of advanced NLP models, adaptive cryptographic techniques with smart contracts on blockchain hence providing an end-to-end solution for securely storing and processing sensitive user data in cloud.

 doi:10.20944/preprints202409.1738.v

Table 1. Related reference findings in our investigation.

Authors	Citation	Objectives	Findings
E. Zeydan, S. S. Arslan and M. Liyanage (2024)	[1]	How current Artificial Intelligence (AI)/Machine Learning (ML) frameworks and available cloud infrastructures in building end-to-end ML lifecycle management for healthcare systems and sensitive biomedical data.	Role of AI and ML for managing life cycle for sensitive user data.
M. Battula (2024)	[2]	Addressing security challenges in Multi-tenant database Management Systems in cloud environment.	Security improvements for the chunks stored in cloud cluster.
K. Sundar, G. Kiran Vishwak and S. G. Eswaran (2024)	[3]	Transformative approach to cloud-based community data sharing, seeking to redefine the dynamics of security and privacy.	Selection of appropriate encryption algorithm for maintaining and managing sensitive data.
O. Layode, H. N. N. Naiho, G. S. Adeleke, E. O. Udeh and T. T. Labake (2024)	[4]	Role of cybersecurity in addressing challenges faced for maintaining sensitive user data.	Use of artificial intelligence, blockchain, and machine learning in enhancing security measures in order to maintain sensitive data.
M. A. Alshammari, H. Hamdi, M. A. Mahmood, and A. A. A. El-Aziz (2024)	[6]	Secure solution for access control in cloud computing environments using blockchain.	By using blockchain technology efficiently, a more secure, scalable, and Transparent access control framework can be implemented.
M. Almasian, A Shafieine-jad (2024)	[7]	Leveraging blockchain technology for secure access control of the user data.	Using blockchain to implement access control as smart contract, wherein user can request to access his file by logging a transaction in the blockchain.
V. G, D. M S, M. Hashmi, J. R. K and K. B V (2024)	[9]	Approach to detect VPN activity and block the user from accessing VPN services using packet sniffing.	Tracking unauthorized user access from the user using VPN service.
Fu, B., Fang, T., Zhang, L., Zhou, Y., and Xiao, H. (2024)	[10]	Combining advanced encryption standard algorithms with elliptic curve cryptography algorithms to generate encryption key pairs through elliptic curve cryptography algorithms.	Use of elliptic curve cryptography to generate key pairs for crypto wallet.

 doi:10.20944/preprints202409.1738.v

Table 1. *Cont.*

Authors	Citation	Objectives	Findings
S. T. Bukhari, M. U. Janjua and J. Qadir, H. (2024)	[11]	Use of elliptic-curve cryptography (ECC) encryption algorithm for storing the seed phrase online by encrypting the seed phrase and using the splitting technique to store the crypto wallet seed phrase.	Generating key pairs for crypto wallets using ECC.
K. G. Babu, J. Naveen, P. V. Vamsi Dhar Reddy, A. Imam and V. S. Vetri Selvi (2023)	[12]	How original IP address can be tracked using Honeypot.	Tracking original IP address of unauthorized access of user data.
G. Sucharitha, V. Sitharamulu, S. N. Mohanty, A. Matta, and D Jose (2023)	[13]	Use of encryption to protect sensitive data.	Usage of Blockchain technology for secure key generation, and for access control while the immutability of the blockchain ensures the confidentiality of ciphertext.
B. Ranganatha Rao, B. Sujatha (2023)	[17]	Key reduction method to make the keys even shorter, which speeds up the Advanced Encryption standard (AES) encryption process.	Usage of ECC algorithm to improve time efficiency of overall system.
S. Khanum and K. Mustafa (2023)	[18]	Sensitive data protection using an air-gapped hardware wallet and transactional privacy by hashing the transaction with the blake3 algorithm.	Protection of sensitive data through hashing and encryption.
M. Kaur and A. B. Kaimal (2023)	[19]	Literature review on detecting cloud computing safety challenges and threats, also offers ideas for resolving data breach issues.	Identification of various security breaches that occurs in cloud environment.
J. Guffey and Y. Li (2023)	[20]	Study on cloud service misconfigurations often lead to massive data leakage or malicious code injection.	How unauthorized access can lead to leakage of sensitive data.

3. Novelty of the Work

This work is novel for its pioneering combination of Natural Language Processing (NLP), encryption, and blockchain-powered smart contracts to establish an overall data security system across cloud environments. Unlike traditional methods that depend on static data classification and basic

 doi:10.20944/preprints202409.1738.v

encryption, this employs advanced NLP techniques like Transformer Models with Named Entity Recognition (NER) to dynamically classify the important sensitive information in a dataset. This automated categorization approach dramatically reduces the likelihood of sensitive data being incorrectly classified, meaning that all essential information are recognized and handled securely. In addition, each data access will use different nonce parameters and every attack causes changes in the dynamic hashing mechanism -> meaning it becomes much harder for attackers to follow or extract Data Patterns, hence greatly improvee system strength. Figure 1 represents the high-level architecture of Management of Crucial Data.

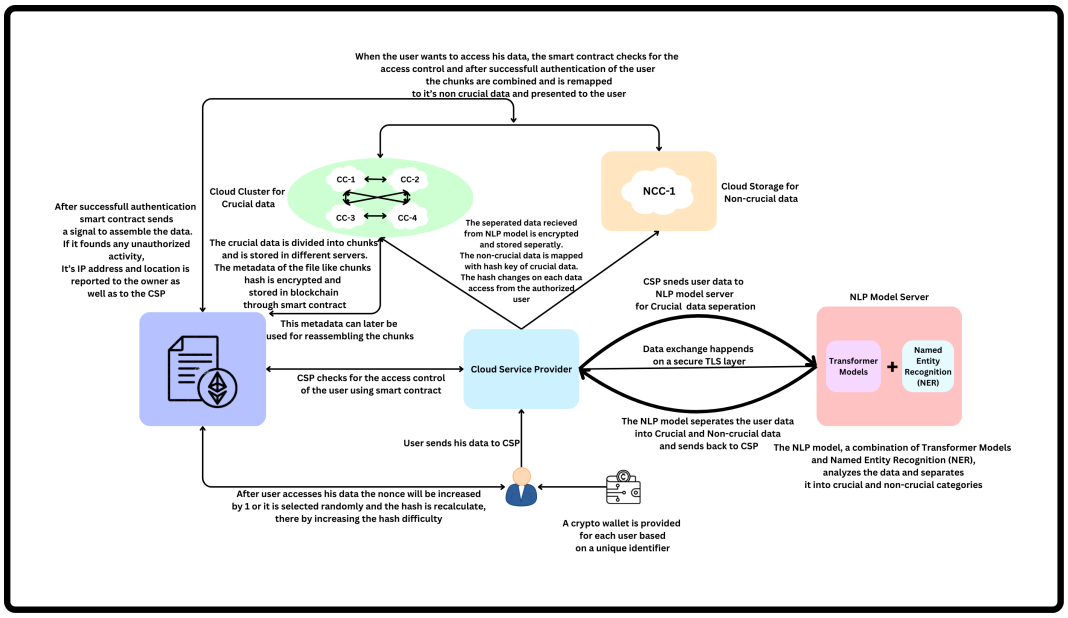

Figure 1. Novel Architecture of Management of Crucial Data.

Moreover, it is this paper that breaks a new ground in employing the blockchain technology and smart contracts for access control and data integrity management. In addition, smart contracts apply automated access protocols that are tamperproof and can immediately identify with high fidelity any unauthorized attempts — creating a log for all suspicious activities which then get reported back to the user as well as the cloud service provider. This also adds a self-managing component that can enhance security measures as various methods of unauthorized access evolve over time. The solution is unique in that it incorporates all of these advanced technologies into a single package, providing an end-to-end alternative method to cloud-based data protection which offers dramatic improvements over current the-state-of-the-art.

4. Secure and Privacy Preserving Crucial Data Management in Cloud Environment.

A mathematical model that formally represents the fundamentals and mechanics of the network is employed as a basis for secure data management in cloud infrastructure. In the case of a database, these include formulas that articulate how data is classified as well as encrypted (and possibly hashed or applied access controls). Using probability distributions and transformer-based architectures, we represent the data classification process in mathematical formulations to identify important from less-important data. We develop a model for encryption and hashing mechanisms to capture the degree of security measures, as well as an exact expression that mathematically defines how adjusting values in nonces dynamically will impact over time data security. Additionally, the smart contract driven access control system is formal-logical and cryptographic proofs are used to give tamper-proof data: they validate who has or had what kind of access rights. Such an approach, with the input of different aspects and knowledge about internal descriptors in various layers provides a theoretical

 doi:10.20944/preprints202409.1738.v

underpinning for this concept also will enable quantitative analysis and optimization that assures framework robustness scale-up to fulfill present security requirements.

4.1. NLP Modelling: Transformer Models with Named Entity Recognition.

NLP component for separating crucial data using Transformer Models with Named Entity Recognition (NER), we need to define the mathematical framework that represents the process of tokenization, attention mechanisms, and the identification of named entities.

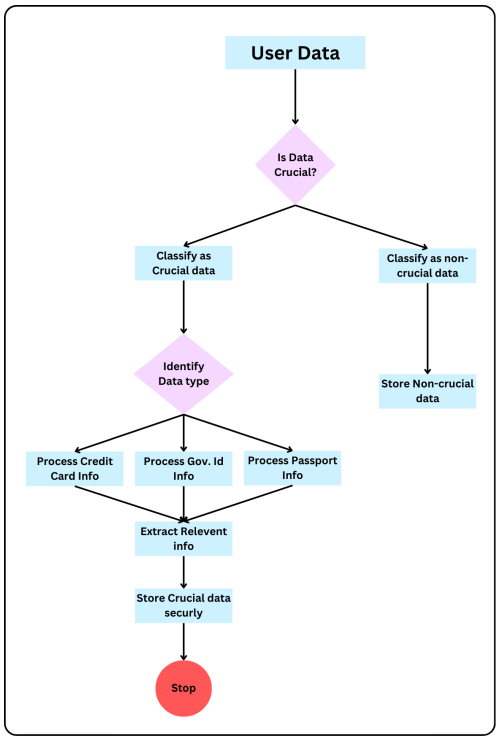

Figure 2. Flow control diagram of NLP model

4.1.1. Variables

- $X = (x_1, x_2, \ldots, x_n)$: Input text sequence of tokens.
- e_i: Embedding vector for token x_i.
- p_i: Positional encoding vector for token x_i.
- Z_i: Combined input representation (embedding + positional encoding).
- Q, K, V: Query, Key, and Value matrices.
- d, d_k, d_{ff}: Dimensions for embeddings, attention heads, and feed-forward layers.
- h: Number of attention heads.
- $H = (h_1, h_2, \ldots, h_n)$: Transformer output representations for each token.
- y_i: Predicted entity label for token x_i.
- c: Number of entity classes.

4.1.2. Input Sequence Representation

Let us define an input text sequence $X = (x_1, x_2, \ldots, x_n)$, where x_i represents the i-th token in the sequence and n is the total number of tokens in the input.

4.1.3. Token Embedding and Positional Encoding

The input sequence X is transformed into embeddings using an embedding matrix E:

$$\text{Embeddings}(X) = E \cdot X = (e_1, e_2, \ldots, e_n) \tag{1}$$

where $e_i \in \mathbb{R}^d$ is the embedding vector for token x_i, and d is the dimensionality of the embeddings.

 doi:10.20944/preprints202409.1738.v

Additionally, we add positional encodings P to capture the order of tokens:

$$Z_i = e_i + p_i, \quad \forall i = 1, \ldots, n \tag{2}$$

where $p_i \in \mathbb{R}^d$ represents the positional encoding vector for the i-th token.

4.1.4. Self-Attention Mechanism

The core of the Transformer model is the self-attention mechanism, which computes the relevance of each token to every other token in the sequence. This is done using three weight matrices: W_Q (query), W_K (key), and W_V (value).

The query Q, key K, and value V matrices for each token are calculated as:

$$Q = Z \cdot W_Q, \quad K = Z \cdot W_K, \quad V = Z \cdot W_V \tag{3}$$

where $Z \in \mathbb{R}^{n \times d}$, and $W_Q, W_K, W_V \in \mathbb{R}^{d \times d_k}$ are learnable parameter matrices, and d_k is the dimension of the queries and keys.

The self-attention score is then computed using the scaled dot-product attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{4}$$

4.1.5. Multi-Head Attention

Multi-head attention allows the model to jointly attend to information from different representation subspaces. Given h attention heads, the output from each head is concatenated and linearly transformed:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W_O \tag{5}$$

where:

$$\text{head}_i = \text{Attention}(Q_i, K_i, V_i), \quad i = 1, \ldots, h \tag{6}$$

and $W_O \in \mathbb{R}^{hd_k \times d}$ is the output weight matrix.

4.1.6. Feed-Forward Neural Network

The output of the multi-head attention is then passed through a position-wise feed-forward network (FFN):

$$\text{FFN}(Z) = \text{ReLU}(ZW_1 + b_1)W_2 + b_2 \tag{7}$$

where $W_1 \in \mathbb{R}^{d \times d_{ff}}, W_2 \in \mathbb{R}^{d_{ff} \times d}$ are learnable parameters, and b_1, b_2 are biases.

4.1.7. Named Entity Recognition (NER)

To perform Named Entity Recognition, the output of the Transformer model $H = (h_1, h_2, \ldots, h_n)$ is fed into a classification layer that predicts the entity label y_i for each token x_i:

$$y_i = \text{softmax}(W_{NER} \cdot h_i + b_{NER}), \quad i = 1, \ldots, n \tag{8}$$

where $W_{NER} \in \mathbb{R}^{d \times c}$ and $b_{NER} \in \mathbb{R}^c$ are the weight matrix and bias for the NER classification, and c is the number of entity classes (e.g., 'CRUCIAL', 'NON-CRUCIAL', 'O').

4.1.8. Loss Function

The training of the NLP model is driven by minimizing the cross-entropy loss for the NER task:

$$\mathcal{L}_{NER} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} y_{i,j}\log(\hat{y}_{i,j}) \tag{9}$$

 doi:10.20944/preprints202409.1738.v

where $y_{i,j}$ is the true entity label, and $\hat{y}_{i,j}$ is the predicted probability for token x_i belonging to class j.

4.2. Crucial Data Storage and Retrieval Modelling.

The mathematical modelling for crucial data storage and retrieval involves using encryption and hashing algorithm. The crucial data encryption is carried out with the public and private key of the crypto wallet provided to the user. Elliptic Curve Cryptography (ECC) algorithm is used for encryption in order to reduce the computation time as we can generate larger key sizes without significantly increasing the key size or CPU and memory requirements. SHA-256 Algorithm is used for hashing the crucial data with the nonce.

4.2.1. Hashing with SHA-256

Given an input of crucial data D and a nonce n, the SHA-256 hashing algorithm is applied to generate a hash value.

- Input: D (Crucial Data), n (Nonce)
- Hash Function:

$$H = \text{SHA-256}(D \| n) \tag{10}$$

where:

- H is the hash output.
- $\|$ denotes concatenation.
- SHA-256 is the secure hash algorithm.

4.2.2. Encrypting with ECC

ECC is used for asymmetric encryption to secure the crucial data.

- Private Key: k_{private}
- Public Key:

$$k_{\text{public}} = k_{\text{private}} \cdot G \tag{11}$$

  where G is a point on the elliptic curve.
- The crucial data D is encrypted with the recipient's public key:

$$C = E(k_{\text{public}}, D) \tag{12}$$

  where:

  - C is the ciphertext.
  - E represents the elliptic curve encryption function.

4.2.3. Splitting and Storing Crucial Data in Chunks

The encrypted crucial data C is split into m chunks:

$$C = (C_1, C_2, \ldots, C_m) \tag{13}$$

Each chunk C_i is stored in a different server within a cloud cluster.

4.2.4. Storing Non-Crucial Data

Non-crucial data N is stored separately on a different cloud server.

4.2.5. Storing Metadata using Smart Contract

Metadata, such as the hash values of the chunks and encryption information, is stored on the blockchain using a smart contract.

 doi:10.20944/preprints202409.1738.v

- Metadata for each chunk:

$$M_i = \text{Metadata}(C_i) = \text{SHA-256}(C_i) \quad \text{for } i = 1, 2, \ldots, m \tag{14}$$

- Smart Contract Execution:

$$\text{SmartContract}.store(M_i) \quad \text{for } i = 1, 2, \ldots, m \tag{15}$$

where M_i represents the metadata stored on the blockchain.

4.2.6. User Access and Access Control Using Smart Contract

When a user requests to access their data:

- Access Request: The user sends a request to access their data to the cloud service.
- Access Control Check: The smart contract checks the access control by verifying the user's credentials and authorization status:

$$\text{SmartContract}.verify(\text{UserID}, \text{AccessRequest}) \tag{16}$$

- Metadata Retrieval: If authorized, the smart contract retrieves the metadata M_i from the blockchain.
- Nonce Update: For security, the nonce n is updated every time the user accesses the data:
  - Nonce Increment Method:

$$n' = n + 1 \tag{17}$$

  - Random Nonce Method:

$$n' = \text{Random}(\text{Range}) \tag{18}$$

4.2.7. Decryption and Data Retrieval

The data is decrypted using the user's private key:

$$D' = D(C, k_{\text{private}}) \tag{19}$$

where:

- D' is the decrypted crucial data.
- D represents the elliptic curve decryption function.

4.3. 8. Changing Hash upon Data Access

After the data is accessed, a new hash is generated using the updated nonce:

$$H' = \text{SHA-256}(D \| n') \tag{20}$$

This new hash makes it challenging for unauthorized parties to track the crucial data.

4.3.1. Summary of Mathematical Equations

- Hashing:

$$H = \text{SHA-256}(D \| n) \tag{21}$$

- Encryption:

$$C = E(k_{\text{public}}, D) \tag{22}$$

- Metadata Storage:

$$M_i = \text{SHA-256}(C_i) \tag{23}$$

 doi:10.20944/preprints202409.1738.v

- Access Control Check:

$$\text{SmartContract}.verify(\text{UserID}, \text{AccessRequest}) \tag{24}$$

- Nonce Update:

$$n' = n + 1 \quad \text{or} \quad n' = \text{Random}(\text{Range}) \tag{25}$$

- Decryption:

$$D' = D(C, k_{\text{private}}) \tag{26}$$

- Hash Update:

$$H' = \text{SHA-256}(D \| n') \tag{27}$$

5. Experimental Results and Discussion

When it comes to protecting and controlling critical data, you need a means of measuring the pros and cons of different cryptographic as well as handling techniques. This testing involves investigating how long it needs for important actions and nonce based hashing strategies to create, obtain or evaluate sensitive data. We consider two nonce strategies: incremental nonce for which we systematically choose the next increased by value and random a randomly chosen from the range. The goal is to explore how these strategies impact hash generation time and the computational load on a node as the hashing difficulty increases. We also investigate how effective it is to use Named Entity Recognition (NER) combined with Transformer models for separating the more important data from blankets of structurally less relevant information. We perform these tests to balance the trade-offs regarding computational efficiency, accuracy, and energy needs of our proposed data management system which is an essential step (also best practice) towards achieving end-to-end security combined with optimal data management operations.

The performance analysis was conducted in a basic local environment using PyCharm to calculate the uploading time and also the time required to generate hash with varying nonce. The last experiment conducted was to calculated the accuracy of the NLP model in two case: 1) Only using Named Entity Recognition and 2) Transformer Models and Named Entity Recognition. The method of implementation was done by deploying the server in 4 different port and one port for cloud cluster (assumption as cloud storage). The performance was conducted on a laptop with specification having AMD Ryzen 7 4800H with Radeon Graphics 2.90 GHz with 16GB RAM of 64-bit operating system, x64-based processor of Windows 11 Operation system.

The findings of the performance evaluation indicated that Transformer models and Named Entity Recognition (NER) together have given better results in detection such essential information as opposed to using NER by itself. NER can extract and categorize named entities with pre-defined labels, but Transformer models bolster this feature by introducing a deeper understanding of context as well the ability to understand intricate relationships in data elements. The attention mechanism present in Transformers enables processing the whole input sequence at once, making fine-grained understanding possible, which is essential for higher accuracy of entity classification. This union brings a combination of the context-aware and pre-trained language capabilities of Transformers to NER for improved detection each time we are dealing with entities that map to important information — like proprietary amounts or personal data including credit card numbers. So, it makes the accuracy of data separation better and stops misclassification that will help to work more effectively in turn with using this combined approach hence builds a secure system for managing our own Data. This integration leads to more robust and secure sensitive data processing, allowing for a better performance than achievable with NER alone.

Figure 3 — it can be seen that the Transformer + NER works a lot slower on trying to separate between propensity and non-propensity data than just using the traditional token-based model. So an even simple example like processing credit card data: using the Transformer + NER model takes about 1020 ms, performing just of only requires 650 ms while is a common pattern even with other

 doi:10.20944/preprints202409.1738.v

types makes sense and seems pretty straight forward — combined ones take more time to process. The longer duration, but with added accuracy of the Transformer model — an acceptable trade off. More importantly, it helps reach a much higher level of accuracy in detecting and labeling sensitive information correctly. Although these procedures take more time, the good news is that much higher accuracy and lower misclassification make it worth waiting as this in-turn demonstrates why both methods are vital to managing and secure important data. Consequently, the much better prediction quality of Transformer+NER model comes at a price in terms of running time but still it is clear that this upgraded version can reliably separate data from logs.

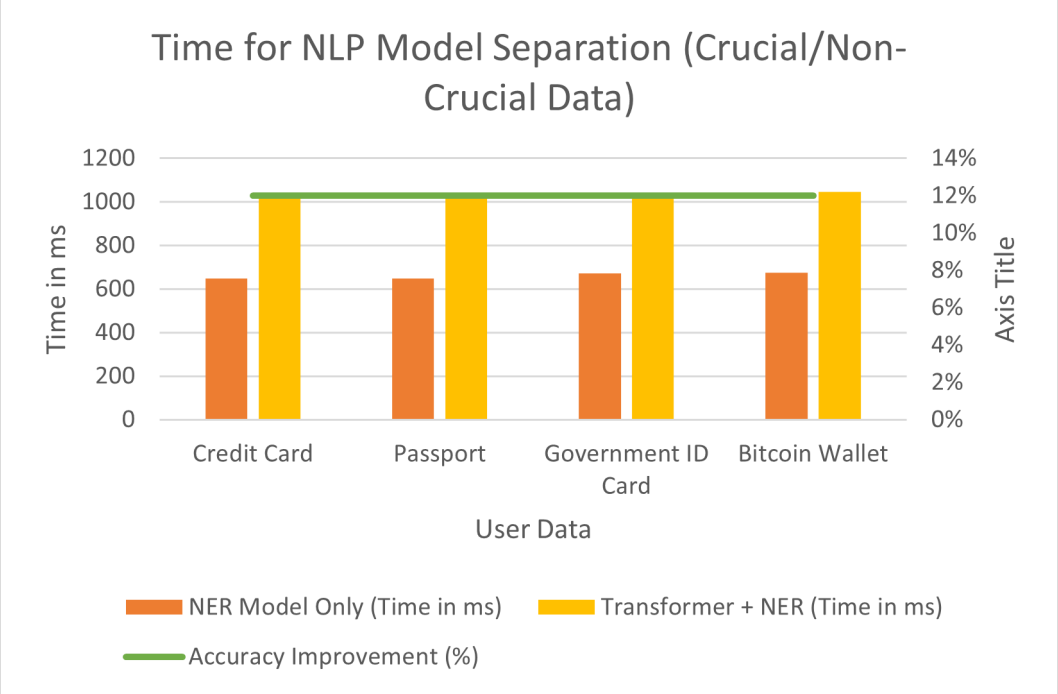

Figure 3. Time for NLP Model Separation of Crucial and Non-Crucial data.

In Figure 4, we can see the percentage of accuracy level turn out when adding a term recognition component to Transformer models in comparison to this model using only its NER. This results in an evident improvement of accuracy when using NER with the Transformer model (see Figure 1). As shown by the Credit card data, combined model has accuracy 98% whereas NER only gives 91% of very low increase. For other data types (passports, government identity card) we see similar improvements: performance increases from 90% with NER to 97%, and it increases from 92% in the case of passports up to almost perfect accuracy (99%). This consistent improvement highlights the effectiveness of context and sophisticated language understanding provided by our transformer model in identifying named entities more accurately. But despite the higher computational requirements of this model, we can see that it provides great benefits in terms of precision and recall for finding sensitive information. It shows the model can reliably separate data more accurately, crucial for strong compliance and security.

 doi:10.20944/preprints202409.1738.v

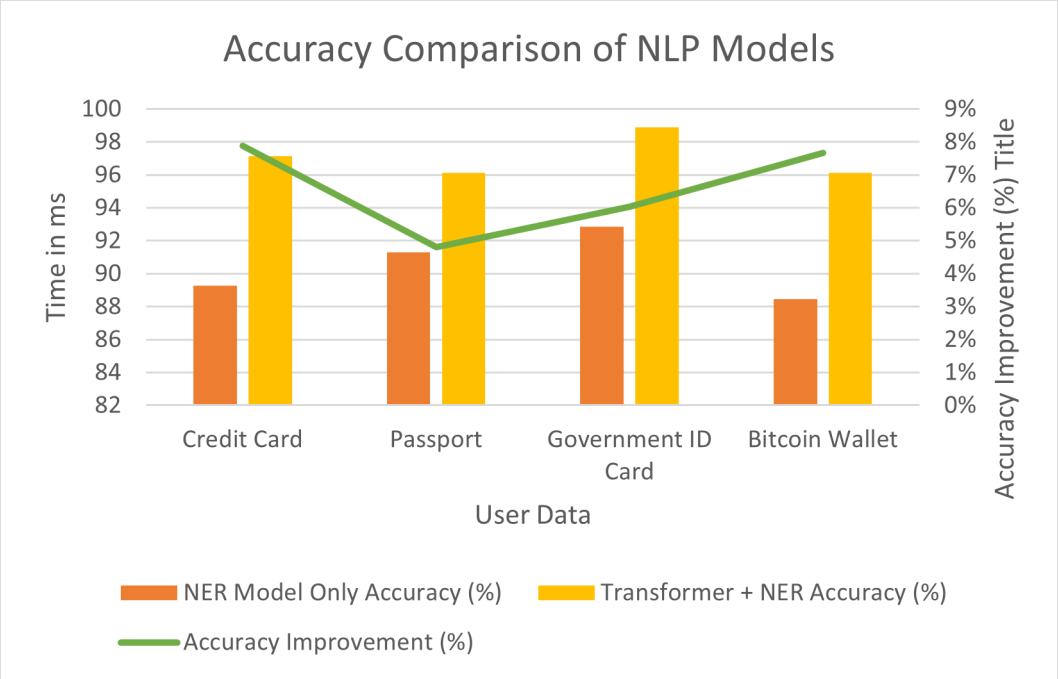

Figure 4. Accuracy Comparison of NLP Models .

Hash generation is essential in cryptographic systems to maintain data integrity and security. This mechanism is directly associated with the nonce, a variable that can be incremented or chosen randomly to modify input data and therefore generate different hash values. In doing so, the hashing process becomes more complex and harder for attackers to predict. The time range for hash generation: The nonce strategy (incremental or random) will have an effect on how long the hashing takes. We can see the computational and energy costs from these two nonce strategies by analyzing in time for hash generation. The next formula and tables (Figures 5 and 6) show the total time needed for hash generation using both Incremental Nonce strategy versus Randomized one, illustrating computational and security tradeoffs of our cryptographic model.

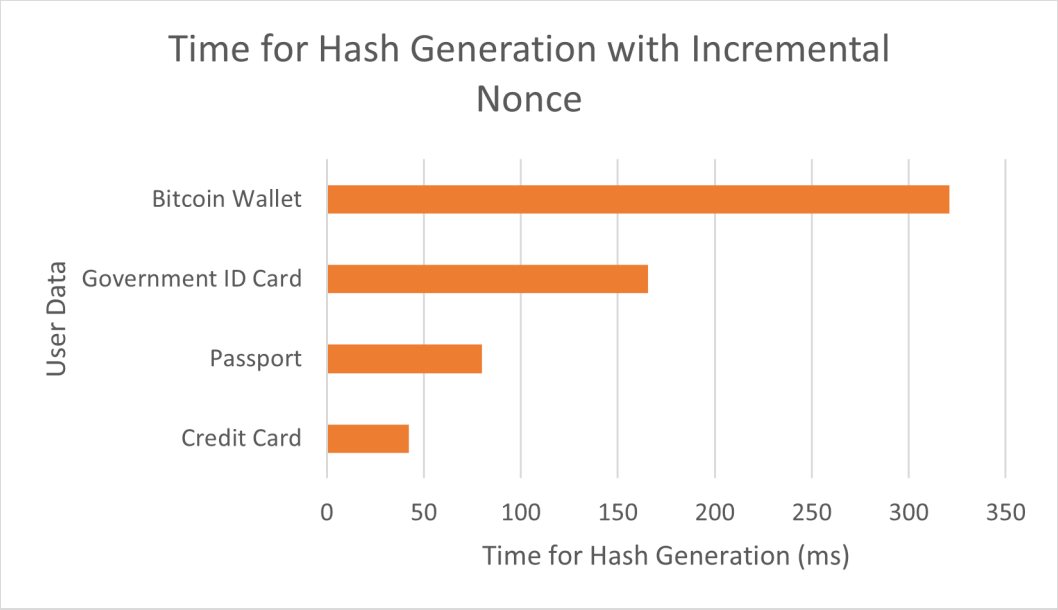

Figure 5. Time for Hash generation with Incremental Nonce.

 doi:10.20944/preprints202409.1738.v

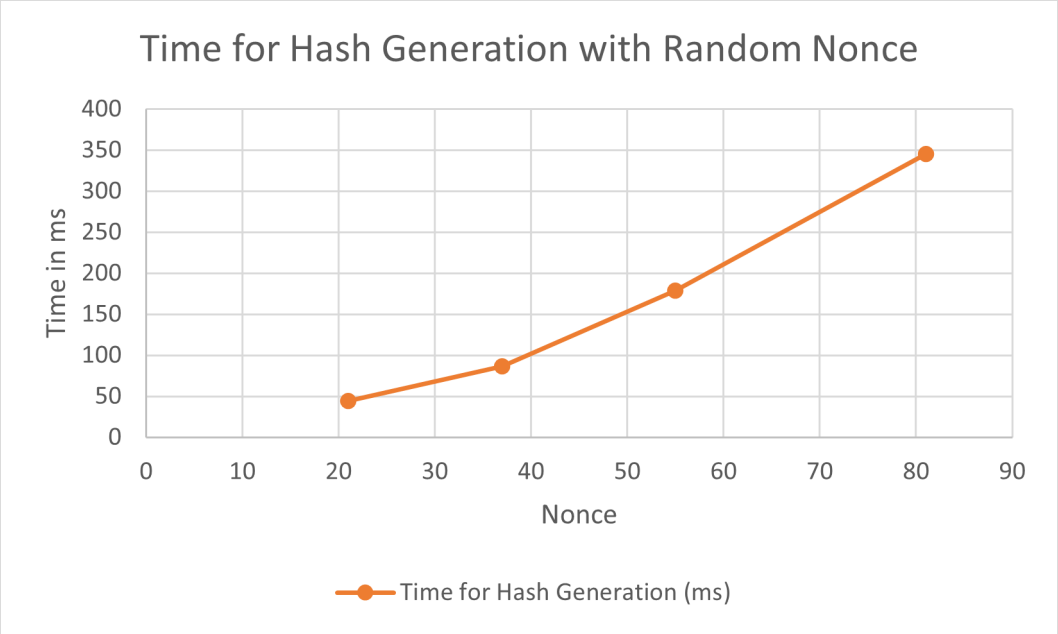

Figure 6. Time for Hash generation with Random Nonce.

5.1. Formula for Hash Generation Based on Nonce

The time for hash generation, $T(n)$, can be represented as a function of the nonce, n, and the difficulty level, D:

$$T(n) = a \cdot 2^{D(n)}$$

where:

- $T(n)$ is the time for hash generation for nonce n.
- $D(n)$ is the difficulty level of the hash for a given nonce n.
- a is a constant representing the base time unit for one hash computation.

The nonce, n, is reset to zero when the following condition is met:

$$D(n) \geq D_{\text{max}} \implies n = 0$$

where:

- D_{max} is the maximum difficulty level, beyond which the computation energy required becomes excessive.

5.2. Time for Hash Generation with Incremental Nonce

Table 2. Time for Hash Generation with Incremental Nonce.

User Data	**Nonce Increment**	**Difficulty Level D**	**Time for Hash Generation $T(n)$ (ms)**
Credit Card	+1 per access	3	42.14
Passport	+1 per access	4	79.87
Government ID Card	+1 per access	5	165.71
Bitcoin Wallet	+1 per access	6	321.11

 doi:10.20944/preprints202409.1738.v

5.3. Time for Hash Generation with Random Nonce

Table 3. Time for Hash Generation with Incremental Nonce.

User Data	Random Nonce n Random between 1-100	Difficulty Level D	Time for Hash Generation $T(n)$ (ms)
Credit Card	21	3	44.12
Passport	37	4	86.44
Government ID Card	55	5	178.62
Bitcoin Wallet	81	6	345.47

5.4. Stopping Condition for Nonce Reset

The nonce, n, will be reset to zero when the hash difficulty level reaches or exceeds the maximum permissible level, $D_{\max}$, to prevent excessive computational energy requirements:

$$\text{If } D(n) \geq D_{\max}, \text{ then } n = 0$$

where:

- $D(n)$ is the current difficulty level associated with the nonce n.
- $D_{\max}$ is the predefined maximum difficulty level threshold.

For instance, when the difficulty level reached is too high $D_{\max} = 6$, then it will reset to zero and keep iterating until a feasible level that doesn't expend more energy than necessary.

This allows the system to balance between data security (via altering hashes) and computational efficiency, modifying the rules on-the-fly depending of hash difficulty.

6. Conclusions

In this study, we have presented a full-frame work for maintaining sensitive user data in the sky by various methods that are Blockchain Technology, modern Natural Language Processing (NLP) models and advanced encryption technologies. Our approach, which integrates Named Entity Recognition (NER) and pre-trained Transformer models classifies critical data sets in a more precise context aware matter for improving the accuracy of managing processes. The model integrates aspects in which existing techniques are typically not very strong (e.g., context understanding, entity disambiguation) but at once delivers a state-of-the-art approach for data identification within the complex and diverse datasets.

In addition, our framework enhances data protection through the use of adaptive cryptographic techniques (e.g., dynamic nonce-based hashing and elliptic curve encryption) designed to protect sensitive information. Our proposed solution is based on storing sensitive data in encrypted chunks distributed over multiple servers and using blockchain with smart contracts to store metadata, that results in more secure, reliable multi-layered protection for integrity of the Data Privacy preserving techniques. Smart contracts allow tamper-proof, automated access control and auditability which reduces the dependence on central authorities to manage processes leading to higher transparency with regulatory compliance.

In total, the proposed methodology is a new fusion of NLP and Encrypted technique with Blockchain technology for good cloud security digital storage. This more than just wraps up such missing areas in an existing research but also introduces a new direction to the camp of secure cloud data management with scalable and non-rigid model, where it can be easily evolved according to any emerging risks or jurisdictions requirements. By adopting this approach, for the first time we demonstrate how clean-slate designs that simultaneously consider both data centric security and privacy can be rigorously designed to kill two birds with one stone in cloud-based big-data management — enabling a new era of secure private public clouds.

 doi:10.20944/preprints202409.1738.v

References

1. E. Zeydan, S. S. Arslan and M. Liyanage, "Managing Distributed Machine Learning Lifecycle for Healthcare Data in the Cloud," in *IEEE Access*, vol. 12, pp. 115750-115774, **2024**, doi: https://doi.org/10.1109/ACCESS.2024.3443520
2. M. Battula, "A Systematic Review on a Multi-tenant Database Management System in Cloud Computing," 2024 *International Conference on Cognitive Robotics and Intelligent Systems (ICC - ROBINS)*, Coimbatore, India, **2024**, pp. 890-897, doi: https://doi.org/10.1109/ICC-ROBINS60238.2024.10533959.
3. K. Sundar, G. Kiran Vishwak and S. G. Eswaran, "Enhancing Cloud Security: Secure and Auditable Data Sharing and its Implementation," 2024 *2nd International Conference on Networking and Communications (ICNWC)*, Chennai, India, **2024**, pp. 1-6, doi: https://doi.org/10.1109/ICNWC60771.2024.10537314.
4. Oluwabunmi Layode, Henry Nwapali Ndidi Naiho, Gbenga Sheriff Adeleke, Ezekiel Onyekachukwu Udeh and Talabi Temitope Labake, "The role of cybersecurity in facilitating sustainable healthcare solutions: Overcoming challenges to protect sensitive data," in *International Medical Science Research Journal*, **2024**, Vol. 4, No. 6, doi: https://doi.org/10.51594/imsrj.v4i6.1228.
5. Akoh Atadoga, Enoch Oluwademilade Sodiya, Uchenna Joseph Umoga and Olukunle Oladipupo Amoo, "A comprehensive review of machine learning's role in enhancing network security and threat detection," in *World Journal of Advanced Research and Reviews*, **2024**, Vol. 23, Issue 3, doi: https://doi.org/10.30574/wjarr.2024.21.2.0501.
6. Alshammari, M.A.; Hamdi, H.; Mahmood, M.A.; El-Aziz, A.A.A. Cloud Computing Access Control Using Blockchain. *Int. J. Intell. Syst. Appl. Eng.* **2024**, *12*, 380–390.
7. Almasian, M.; Shafieinejad, A. Secure cloud file sharing scheme using blockchain and attribute-based encryption. *Comput. Stand. Interface* **2024**, *87*, 103745. https://doi.org/10.1016/j.csi.2023.103745.
8. Hamid, I.; Frikha, M. Blockchain-Enhanced Cybersecurity and Privacy in Cloud Computing: A Systematic Literature Review. *J. Theor. Appl. Inf. Technol.* **2024**, *102*, 514–531.
9. V. G, D. M S, M. Hashmi, J. R. K and K. B V, "Robust Technique for Detecting and Blocking of VPN over Networks," *Ninth International Conference on Science Technology Engineering and Mathematics (ICONSTEM), Chennai, India*, **2024**, pp. 1-5, doi: https://doi.org/10.1109/ICONSTEM60960.2024.10568824.
10. Fu, B., Fang, T., Zhang, L., Zhou, Y., and Xiao, H., "Communication security of intelligent information service platform combining AES and ECC algorithms", In *Journal of Cyber Security Technology*, **2024**, 1–18, doi: https://doi.org/10.1080/23742917.2024.2371053
11. S. T. Bukhari, M. U. Janjua and J. Qadir, "Secure Storage of Crypto Wallet Seed Phrase Using ECC and Splitting Technique," in *IEEE Open Journal of the Computer Society*, vol. 5, pp. 278-289, **2024**, doi: https://doi.org/10.1109/OJCS.2024.3398794.
12. K. G. Babu, J. Naveen, P. V. Vamsi Dhar Reddy, A. Imam and V. S. Vetri Selvi, "Tracing Phishing Website Original IP Address," *International Conference on Networking and Communications (ICNWC)*, Chennai, India, **2023**, pp. 1-5, doi: https://doi.org/10.1109/ICNWC57852.2023.10127555.
13. Sucharitha, G.; Sitharamulu, V.; Mohanty, S.N.; Matta, A.; Jose, D. Enhancing Secure Communication in the Cloud Through Blockchain Assisted-CP-DABE. *IEEE Xplore* **2023**, *11*, 99005–99015, doi: https://doi.org/10.1109/ACCESS.2023.3312609.
14. Dubey, H.; Roy, K. Secure Access Control in Cloud Computing Environments: Smart Contract Blockchain. *Vidhyayana* **2023**, *8*, 392–404.
15. Prasad, S.N.; Rekha, C. Block chain based IAS protocol to enhance security and privacy in cloud computing. *Meas. Sens.* **2023**, *28*, 100813, doi: https://doi.org/10.1016/j.measen.2023.100813.
16. Pansara, R., "Navigating Data Management in the Cloud - Exploring Limitations and Opportunities", In *Transactions on Latest Trends in IoT*, **2023**, 6(6), 57-66.
17. B. Ranganatha Rao, B. Sujatha, "A hybrid elliptic curve cryptography (HECC) technique for fast encryption of data for public cloud security", In *Measurement: Sensors*, Vol. 29, **2023**, 100870, ISSN 2665-9174, doi: https://doi.org/10.1016/j.measen.2023.100870.
18. S. Khanum and K. Mustafa, "Sensitive Data Protection at Blockchain Endpoints and Transactions," In *OPJU International Technology Conference on Emerging Technologies for Sustainable Development (OTCON), Raigarh, Chhattisgarh, India*, **2023**, pp. 1-6, doi: https://doi.org/10.1109/OTCON56053.2023.10113931.

19. M. Kaur and A. B. Kaimal, "Analysis of Cloud Computing Security Challenges and Threats for Resolving Data Breach Issues," *International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India*, **2023**, pp. 1-6, doi: https://doi.org/10.1109/ICCCI56745.2023.10128329.
20. J. Guffey and Y. Li, "Cloud Service Misconfigurations: Emerging Threats, Enterprise Data Breaches and Solutions," In *IEEE 13th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA*, **2023**, pp. 0806-0812, doi: https://doi.org/10.1109/CCWC57344.2023.10099296.
21. Rajguru, S.N.; Choubey, S.K. Blockchain in Cloud Computing for Securing Documents. *Int. Res. J. Mod. Eng. Technol. Sci. (IRJMETS)* **2023**, *5*, 123–130. https://www.doi.org/10.56726/IRJMETS38879.
22. Gousteris, Solonas, Yannis C. Stamatiou, Constantinos Halkiopoulos, Hera Antonopoulou, and Nikos Kostopoulos. "Secure distributed cloud storage based on the blockchain technology and smart contracts." *Emerging Science Journal 7*, **2023**, no. 2, pp. 469-479.
23. Alsuwat, W.; Alsuwat, H. A Survey on Cloud Storage System Security via Encryption Mechanisms. *Int. J. Comput. Sci. Netw. Secur.* **2022**, *22*, 52–61.
24. Mandal, S.; Khan, D.A.; Jain, S. Cloud-Based Zero Trust Access Control Policy: An Approach to Support Work-from-Home Driven by COVID-19 Pandemic. *New Gener. Comput.* **2021**, *39*, 599–622. https://www.doi.org/10.1007/S00354-021-00130-6.
25. Mahmood, G.S.; Huang, D.J.; Jaleel, B.A. A Secure Cloud Computing System by Using Encryption and Access Control Model. *J. Inf. Process. Syst.* **2019**, *15*, 538–549. https://doi.org/10.3745/JIPS.03.0117.