THE SCIENTIFIC TEMPER (2026) Vol. 17 (4): 6005-6009

E-ISSN: 2231-6396, ISSN: 0976-8653

Doi: 10.58414/SCIENTIFICTEMPER.2026.17.4.09

https://scientifictemper.com/

RESEARCH ARTICLE

Enhanced Positional Vigenère (EPV): A Confidentiality-Enabled Encryption Technique for Secure Cloud Storage

B. Fathimamary[1*], M. Baby Nirmala[2], M. Nasreen[3]

Abstract

Cloud computing is an internet-based computing paradigm that provides various hosting and delivery services over the internet. It offers computational resources to users based on their demand. Data storage is one of the main benefits of cloud computing. It provides users with plenty of space to store their data neatly and access it easily, no fuss involved. More and more companies are jumping on cloud platforms that offer Storage as a Service (STaaS), helping them skip the hefty upfront costs and ongoing maintenance of their own servers. However, when organizational and enterprise data are moved to public cloud storage, ensuring data protection and security becomes a critical concern. If unencrypted data is transmitted to the public cloud, there is a possibility of data breaches during transmission. To address this issue, an efficient technique called Enhanced Positional Vigenère (EPV) is proposed in this work. Strengthening the ciphertext and improving data security are the goals of the suggested approach. The EPV technique is implemented in Java, and our experiments show it boosts performance, ramps up efficiency, and makes the ciphertext even more complex.

Keywords: Cloud Computing, Data Security, Encryption, Enhanced Positional Vigenère (EPV), Cloud Storage.

Introduction

Cloud computing uses the internet to deliver hardware and software services. The hardware, storage, networks, interfaces, and services that make up the "cloud" enable users to access infrastructures, processing power, applications, and services whenever they want, regardless of where they are located [thabit et al., 2022]. Three types of services are offered by the cloud computing concept, including Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS), all of which refer to a service-oriented architecture. In that Infrastructure as a Service, *Server, Storage and Network* services are provided by the Cloud. Cloud computing shines brightest for data storage. Cloud Storage provides environment to store their use data via Storage as a Services for cloud users. Because of these benefits each and every organization moves their data to the cloud[sana et al., 2023].

Besides the cloud benefits, it has number of issues related to security. Security tops the list of concerns in cloud environments. Once the user data sent to the cloud, cloud storage provider(CSP)only responsible for that user data. The cloud storage is fully outsourced to the Cloud Service Provider (CSP). That is, such users can't get their hands on their data or see where it's stored for themselves, and they typically have no idea of its exact whereabouts. Cloud is a public environment, where there is lot of possibilities to attack and hack the user data. Therefore, it is necessary to safeguard that data from unwanted access. Cryptography is used in cloud data security to ensure data protection. These days, cryptography is thought of as a combination of three different kinds of algorithms. Symmetric-key algorithms, asymmetric-key algorithms, and hash function algorithms are the three types of algorithms.

[1]Assistant Professor, Department of Commerce Computer Applications, St.Joseph's College (Autonomous), Tiruchirappalli, India

[2]Associate Professor, Department of Computer Science and Engineering, Dhanalakshmi Srinivasan University, Tiruchirappalli, India

[3]Assistant Professor, Department of B.Voc. Software Development, Holy Cross College(Autonomous), Tiruchirappalli, India

***Corresponding Author:** B. Fathimamary, Assistant Professor, Department of Commerce Computer Applications, St.Joseph's College (Autonomous), Tiruchirappalli, India, E-Mail: fathimamary_cc2@mail.sjctni.edu

How to cite this article: Fathimamary, B., Nirmala, M.B., Nasreen, M. (2026). Enhanced Positional Vigenère (EPV): A Confidentiality-Enabled Encryption Technique for Secure Cloud Storage. The Scientific Temper, **17**(4):6005-6009.

Doi: 10.58414/SCIENTIFICTEMPER.2026.17.4.09

Source of support: Nil

Conflict of interest: None.

Received: 25/03/2026 **Accepted:** 08/04/2026 **Published:** 25/04/2026

Cryptography is the study of encrypting and decrypting data. There are numerous cryptography methods for both encryption and decryption. Conventional and public key cryptography are two categories of cryptographic methods [gupta et al.,2023]. Another name for conventional cryptography is symmetric key cryptography. Asymmetric (or public key) cryptography employs a public key for encryption and a private key for decryption, in contrast to symmetric key cryptography, which uses a single key for both. Tim Mather et al.,2009 claims that symmetric encryption is more appropriate for handling encryption in the shortest amount of time and effective for handling massive amounts of data in cloud storage.

Data security is one of the main issues with cloud storage. Users of cloud computing frequently have little control and monitoring power over their data which may be spread across multiple data centers. When users transfer their data to public cloud storage security risks related to data transmission may become apparent. In order to improve data security, user-side encryption techniques must be put into place. This study suggests a confidentiality-enabled method that ensures safe data transmission and storage by encrypting data prior to its transfer to cloud storage.

Related Works

The proposed technique is inspired by previous works in which different methodologies were employed to secure data stored in the cloud using cryptographic techniques. Some of the related works are reviewed as follows:

To handle encryption and decryption procedures and safeguard data kept in cloud environments Jagadeeshwar et al.,2024 suggests an automated security model based on cryptography. Patil et al., 2024 enhanced privacy protection in cloud computing by strengthening Ciphertext-Policy Attribute-Based Encryption with hashing and signature validation techniques. Rupa et al., 2023 introduced a novel homomorphic encryption scheme by means of matrix transformations to achieve security of cloud data utilizing symmetric keys. Saginayev et al.,2024 suggested a hybrid cryptography system based on AES-256, RSA and SHA-256 for the protection of cloud file storage.

Truong et al.,2025 made an extensive survey on security issues of distributed infrastructures such as data confidentiality in the multi-clouds, identity and access management (IAM), emerging cross-cloud threats. These researchers additionally elaborated on privacy preserving techniques and cryptographic protections for multi-cloud environments complex multicloud ecosystems.

AI-based security is increasingly becoming popular for real-time threat detection and automatic incident response in cloud environments. Shaffi et al.,2025 studied the application of ML and predictive analytics for anomaly detection and response plan, concluding that AI can greatly improve the cloud resilience to changing cyber-attacks albeit with ethical and reliability issues. Cloud IoT security related research also has impact on data security in distributed environment. Pathak et al.,2024 reviewed threat and security mechanisms at the edge, cloud-IoT integrated, and cloud with the support of blockchain, ML and some other emerging technologies and highlighted the need for cohesive multi-layer defense solutions to maintain privacy and to avoid data exfiltration.

Enhanced Positional Vigenère (EPV) Technique

This EPV technique is proposed to overcome existing limitations and to preserve the privacy and confidentiality of data stored in the cloud. The objective of the system is to encrypt the given plaintext using a key selected by the sender and to decrypt the ciphertext using the same key. Before encryption, repeated letters in the plaintext are removed and stored separately, thereby reducing the size of the ciphertext. After decryption, the previously removed repeated letters are reinserted into the recovered plaintext to reconstruct the original message. The proposed work is implemented using a simple mathematical formula. Encryption is performed by adding the positional values of the plaintext and the key, while decryption is carried out by subtracting the key values from the ciphertext. However, the key value must be smaller than the plaintext and ciphertext values to ensure proper computation.

Furthermore, the proposed technique is extended to support both uppercase and lowercase alphabets. It removes repeated alphabets and spaces from the plaintext, and their positions are stored separately. In this approach, 26 uppercase and 26 lowercase alphabets are used, resulting in a total of 52 characters. Equation 1 is used for encryption is:

$$\text{Ciphertext} = (\text{Plaintext} + \text{Key}) \bmod 52 \quad (1)$$

Equation 1 is used for decrypting the ciphertext is:

$$\text{Plaintext} = (\text{Ciphertext} - \text{Key}) \bmod 52 \quad (2)$$

Each uppercase and lowercase alphabet is assigned a unique positional value. These positional values are substituted into the formula for the given plaintext and key. The 52 alphabets are arranged in a predefined order, and the positional values assigned to them are shown in the figure 1 below.

Illustration of the EPV Technique

Encryption Phase:

Step 1: The plaintext "Be Positive" is selected as the input message to be encrypted.

Step 2: The repeated letters and spaces are removed from the plaintext, and the positions of those letters are stored separately. The resulting processed plaintext becomes "BePositv."

Step 3: The plaintext is divided into separate blocks, and positional values are assigned to the letters based on the positional Mapping Matrix.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25
a	b	c	d	e	f	g	h	i	j	k	l	m
26	27	28	29	30	31	32	33	34	35	36	37	38
n	o	p	q	r	s	t	u	v	w	x	y	z
39	40	41	42	43	44	45	46	47	48	49	50	51

Figure 1: positional Mapping Matrix

B e P o s i t v
1 30 15 40 44 34 45 47

Step 4: Select the encryption key. In this illustration, the selected key is "DEAD."

Step 5: Divide the key into blocks corresponding to the plaintext blocks and assign positional values to each letter of the key, as shown below:

Key:
D E A D D E A D
Values:
3 4 0 3 3 4 0 3

Step 6: Apply the positional values of the plaintext and the key sequentially in the encryption formula 1:

Ciphertext = (Plaintext + Key) mod 52

For example,

Ciphertext = (1 + 3) mod 52 = 4

The resulting value is 4. Referring to the value table, the letter corresponding to position 4 is E. Therefore, the ciphertext character is E.

Step 7: Repeat Step 6 for the remaining plaintext and key values. The resulting ciphertext is **"EiPrvmty."**

Decryption Phase

Step 1: Obtain the ciphertext, the key, and the position file.

Step 2: Divide the ciphertext into separate blocks and assign positional values to each letter according to the value table, as shown below:

E i P r v m t y
4 34 15 43 47 38 45 50

Step 3: Divide the key into blocks corresponding to the ciphertext blocks and assign positional values to each letter of the key, as shown below:

D E A D D E A D
3 4 0 3 3 4 0 3

Step 4: Apply the positional values of the ciphertext and the key sequentially in the decryption formula 2:

Plaintext = (Ciphertext – Key) mod 52

For example,

Plaintext = (4 – 3) mod 52= 1

The resulting value is 1. Referring to the value table, the letter corresponding to position 1 is B. Therefore, the decrypted plaintext character is B.

Step 5: Repeat Step 4 for the remaining ciphertext and key values. The resulting plaintext is **"BePositv."**

Step 6: Using the given position file, arrange the obtained plaintext in the original order. The reconstructed plaintext is **"Be Positive."**

Results and Discussion

The Java programming language is used to implement the suggested technique. The output of the proposed framework is described in this section. The plaintext to be encrypted is entered and saved in the designated plaintext notepad as shown in the Figure 2. Similarly, the key used for encryption is entered and saved in the designated key notepad, as shown in the Figure 3.

After setting the path, the encryption and decryption processes are executed, and finally the original plaintext is obtained, as shown in the Figure 4.

As a result, the given key, plaintext, and the positions of the previously removed repeated letters and spaces are displayed. The encrypted text and the decrypted text generated using the key are also displayed along with the restored repeated letters and spaces. In this example, the plaintext is "Be Positive" and the key is "DEAD." The resulting ciphertext is "EiPrvmty." After decryption using the same key, the original plaintext is successfully recovered. Thus, the plaintext is accurately reconstructed after decrypting the ciphertext with the corresponding key. This enhanced approach increases the complexity of the classical Vigenère cipher and improves the security of cloud-stored data, making it more resistant to unauthorized access and cryptanalysis.

Features of the Proposed EPV Technique

Cloud computing faces various challenges in storing users' data securely. The proposed technique helps manage cloud data while preserving its confidentiality. This technique includes several important features, which are described as follows:

- The proposed technique provides enhanced security compared to existing algorithms. It is extended to support both uppercase and lowercase letters, and the positions of the alphabets and spaces are stored separately.
- Each alphabet is assigned a positional value, and this position is changed dynamically during the encryption process. The encryption process is performed only on unique letters, excluding repeated letters and spaces.
- By eliminating repeated letters and spaces, the file size is reduced, thereby minimizing storage requirements. The decryption process can be performed only when both

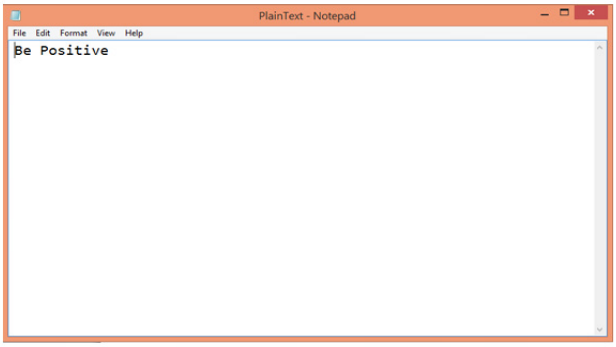

Figure 2: plain text

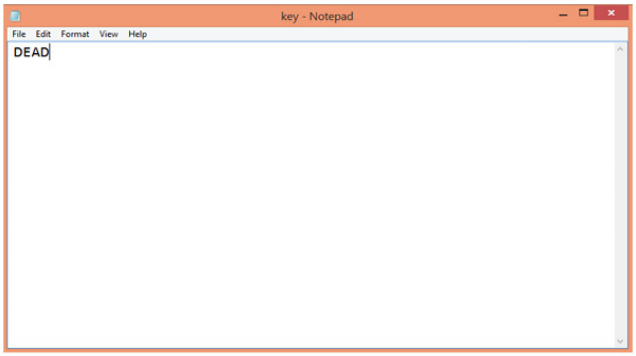

Figure 3: Key

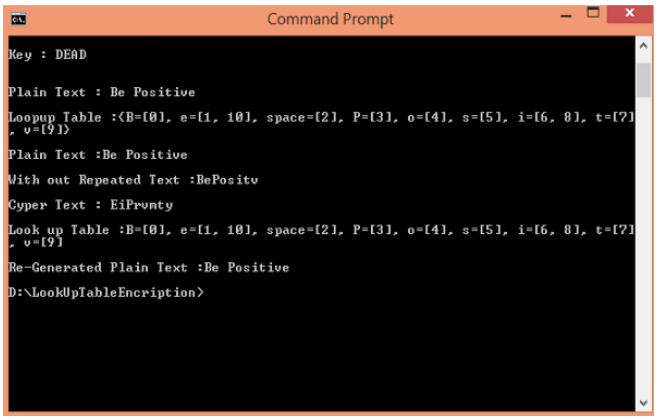

Figure 4: output file

the key and the position file are provided to the receiver.

- This technique is suitable for both large and small data sizes, as it reduces the size of the encrypted data file. Consequently, the file can fit into limited memory space.

Conclusion

This paper proposes a novel approach to cloud computing data security problems. In day-to-day life, cloud computing plays a significant role, as it provides a platform for storing and managing data over the Internet as an alternative to a computer's local memory or hard drive. The objective of this work is to ensure secure data sharing in the cloud. This EPV technique is inspired by the classical Vigenère cipher, which was originally implemented only for uppercase letters. The proposed method extends the algorithm to support both uppercase and lowercase letters by applying a simple mathematical formula. Additionally, repeated letters and spaces are removed during the encryption process and are reinserted during decryption to reconstruct the original plaintext. Experimental results demonstrate that the proposed technique provides enhanced security compared to traditional approaches. The method involves simple steps for encryption and decryption, making the system easy to operate and user-friendly. Future work includes extending the system to support numerical values and special characters. The proposed system can be implemented in both public and private cloud environments.

Acknowledgement

The First author thank, DST-FIST, Government of India for funding towards infrastructure facilities at St.Joseph's College (Autonomous), Tiruchirappalli- 620 002.

References

Amalarethinam, D. I. G., & Fathima Mary, B. (2015a). DMUCE – A confidentiality enabled technique to improve cloud user security. *International Journal of Applied Engineering Research, 10*(14), 34103–34108. https://www.ripublication.com/ijaer.html

Amalarethinam, D. I. G., & Fathima Mary, B. (2015b). eDSSuMRT – Ensured data security strategy using matrix random traversal in cloud storage environment. *International Journal of Applied Engineering Research, 10*(Special Issue 82), 272–279. https://www.ripublication.com/Volume/ijaerv10n82spl.html

Amalarethinam, D. I. G., & Fathima Mary, B. (2016). Confidentiality technique for enhancing data security in public cloud storage using data obfuscation. *International Journal of Control Theory and Applications*, 1–11.

Amalarethinam, D. I. G., & Fathima Mary, B. (2017). Data security enhancement in public cloud storage. *Journal of Computing and Intelligent Systems*, 1–5. https://shcpub.edu.in/web/binary/view_document/?id=1799&model=ir.attachment

Amalarethinam, D. I. G., & Fathima Mary, B. (2018). Security enhancement for public cloud storage with minimum cost. *International Journal of Pure and Applied Mathematics*, 1–9. http://www.ijpam.eu/contents/2018-118-6/36/36.pdf

Amalarethinam, D. I. G., & Fathima Mary, B. (2019). Comparative analysis of obfuscation techniques in public cloud. *American International Journal of Research in Science, Technology, Engineering & Mathematics*, 309–312.

Amalarethinam, D. I. G., & Fathimamary, B. (2016). Data security enhancement in public cloud storage using data obfuscation and steganography. In *Proceedings of the World Congress on Computing and Communication Technologies (WCCCT)* (pp. 181–184). IEEE. https://doi.org/10.1109/WCCCT.2016.41

Amalarethinam, D. I. G., & Vinnarasi, J. (2024). Enhancing the data security of cloud computing critical systems using layered encryption technique. *International Journal of Critical Computer-Based Systems*. https://doi.org/10.1504/IJCCBS.2024.143209

Fathimamary, B. (2025a). A comprehensive study on cryptographic

approaches for securing data in cloud storage. *International Multidisciplinary Research Journal Reviews, 2*(11), 66–72. https://doi.org/10.17148/IMRJR.2025.021107

Fathimamary, B. (2025b). Data security enhancement in cloud using magic square obfuscation. *International Multidisciplinary Research Journal Reviews, 2*, 66–70. https://doi.org/10.17148/IMRJR.2025.020908

Fathimamary, B. (2026a). A novel magical encryption technique for secure and efficient public cloud storage. *Advancement of Computer Technology and Its Applications, 9*(2), 45–50. https://doi.org/10.5281/zenodo.18397099

Fathimamary, B., Nasreen, M., & Velusamy, K. (2026). An efficient DDoS attack detection using optimized long short-term optimization based on improved brainstorm optimization. *Indian Journal of Science and Technology, 19*(5), 298–312. https://doi.org/10.17485/IJST/v19i5.2020

Gupta, M., Ahuja, L., & Seth, A. (2023). Security enhancement in a cloud environment using a hybrid chaotic algorithm with multifactor verification for user authentication. *International Journal of Computers and Applications*. https://doi.org/10.1080/1206212X.2023.2250014

Jagadeeshwar, M., Shanthi, D., & Prasad, M. (2024). Automated data security model using cryptography techniques in cloud environment. *International Journal of Intelligent Systems and Applications in Engineering, 12*(3), 1910–1917.

Mather, T., Kumaraswamy, S., & Shahed, L. (2009). Chapter title. In *Cloud security and privacy* (pp. 61–71). O'Reilly Media.

Pathak, M., Mishra, K. N., & Singh, S. P. (2024). Securing data and preserving privacy in cloud IoT-based technologies: An analysis of threats and safeguards. *Artificial Intelligence Review, 57*, 269.

Patil, S. P., Basthikodi, M., Kumaraswamy, S., Gurpur, A. P., & Raga, A. (2024). Enhancing data privacy protection in cloud computing through ciphertext-policy attribute-based encryption. *Journal of Electrical Systems, 20*(3s).

Pothireddy, S., & Peddisetty, N. (2023). Data security in cloud environment by using hybrid encryption technique: A comprehensive study on enhancing confidentiality and reliability. *International Journal of Intelligent Engineering and Systems*. https://doi.org/10.22266/ijies2023.0630.15

Rupa, C., Greeshmanth, & Shah, M. A. (2023). Novel secure data protection scheme using Martino homomorphic encryption. *Journal of Cloud Computing, 12*, 47.

Saginayev, E., Niyazov, A., Razaque, A., Kalpeyeva, Z., & Urazgaliyeva, A. (2024). Secure file storage on cloud using hybrid cryptography. *Computing & Engineering, 2*(2).

Sana, M. U., Li, Z., Kiren, T., Liaqat, H. B., Naseem, S., & Saeed, A. (2023). A secure method for data storage and transmission in sustainable cloud computing. *Computers, Materials & Continua, 75*(2), 2741–2757. https://doi.org/10.32604/cmc.2023.036093

Shaffi, S. M., Vengathattil, S., Sidhick, J. N., & Vijayan, R. (2025). AI-driven security in cloud computing: Enhancing threat detection, automated response, and cyber resilience [Preprint]. *arXiv*.

Thabit, F., Can, O., Alhomdy, S., Al-Gaphari, G. H., & Jagtap, S. (2022). A novel effective lightweight homomorphic cryptographic algorithm for data security in cloud computing. *International Journal of Intelligent Networks, 3*, 16–30. https://doi.org/10.1016/j.ijin.2022.04.001

Tripathi, D. R. (2020). Data obfuscation technique for security in cloud computing. *International Journal of Recent Technology and Engineering, 8*(5), 4239–4244. https://doi.org/10.35940/ijrte.E6720.018520

Truong, H. L., Nguyen, D. H., et al. (2025). Security and privacy in multi-cloud and hybrid cloud environments: Challenges, strategies, and future directions. *Computers & Security, 157*, 104599.