Ciphertext-Only Attack on a Secure k-NN Computation on Cloud

Shyam Murthy
IISc Bangalore, IN
shyamsm1@iisc.ac.in

Santosh Kumar Upadhyaya
IIIT Bangalore, IN
santosh.upadhyaya@iiitb.ac.in

Srinivas Vivek
IIIT Bangalore, IN
srinivas.vivek@iitb.ac.in

Abstract—The rise of cloud computing has spurred a trend of transferring data storage and computational tasks to the cloud. To protect confidential information such as customer data and business details, it is essential to encrypt this sensitive data before cloud storage. Implementing encryption can prevent unauthorized access, data breaches, and the resultant financial loss, reputation damage, and legal issues. Moreover, to facilitate the execution of data mining algorithms on the cloud-stored data, the encryption needs to be compatible with domain computation. The k-nearest neighbor (k-NN) computation for a specific query vector is widely used in fields like location-based services. Sanyashi et al. (ICISS 2023) proposed an encryption scheme to facilitate privacy-preserving k-NN computation on the cloud by utilizing Asymmetric Scalar-Product-Preserving Encryption (ASPE). In this work, we identify a significant vulnerability in the aforementioned encryption scheme of Sanyashi et al. Specifically, we give an efficient algorithm and also empirically demonstrate that their encryption scheme is vulnerable to the ciphertext-only attack (COA).

Index Terms—Cloud Computing, Cryptanalysis, k-NN, Privacy, Ciphertext-Only Attack

I. INTRODUCTION

As cloud computing continues to evolve, an increasing number of data owners (DO) are transferring their data to the cloud [10] [11]. This shift aids DOs in alleviating the burden associated with data management, computation, and query processing [1] [8]. However, the move towards cloud services also raises concerns regarding data security and privacy. Thus, the choice of encryption protocol becomes crucial, especially when computations need to be performed on encrypted data. Traditional encryption techniques, while effective for securing data, do not offer the capability to perform computational operations within the encrypted domain. Traditionally, in order to perform computations on data, the data must first be decrypted, potentially exposing it to security risks. On the other hand, homomorphic encryption schemes [5] [6] [7] provide a potential solution to this issue. These schemes are designed to allow computations to be carried out directly on encrypted data, without the need for decryption. However, despite these potential benefits, the effectiveness of homomorphic encryption schemes for real-world data computations is not entirely certain. These schemes can be complex and computationally intensive, which can limit their efficiency and practicality for large-scale or real-time data computations. Ideally, encryption schemes that secure the data as well as support search processing

would be best suited for this scenario [13] [14] [15] [16]. Computing k number of nearest neigbours (k-NN) of a given query point in a database, according to some metric, is an important technique in the field of machine learning, among others. Privacy-preserving k-NN is, hence, equally important in privacy-preserving machine learning (PPML). Wong et al. [2] introduced Asymmetric Scalar-product-Preserving Encryption (ASPE), a scheme that preserves the scalar product ordering between two encrypted data points, when searching an ASPE encrypted database. While the work of Zhu et al. [3] encrypted the queries using the Paillier scheme, the work of Sanyashi et al. [4] encrypts the queries in the ASPE scheme itself thereby making the query encryption and the overall scheme more efficient. In the scheme of Sanyashi et al. [4], encrypting a data tuple involves affine-shifting the individual data items by a secret vector, appending a vector of random nonces, and then multiplying the resulting vector by a secret random matrix results in a ciphertext. The authors argue that such a product vector serving as a ciphertext exhibits a high degree of randomness thereby preserving its security.

Prior Attacks on the ASPE scheme: Chunsheng et al. [17] gave a known-plaintext attack on the original ASPE scheme by solving the ciphertext equations corresponding to known plaintexts. Li et al. [12] used independent component analysis (ICA), which is used for blind source separation in signal processing to show that the original ASPE scheme is not secure against ciphertext-only attacks (COA). In this work, we present a COA attack on the scheme of Sanyashi et al. [4], where we make use of linear independence properties of differences of encryptions of two data tuples to distinguish between two sets of ASPE-like encrypted ciphertexts (Sec. III). We stress that the attacks on the original ASPE scheme does *not* necessarily imply attacks on the scheme of Sanyashi et al. [4] due to ciphertext randomization.

A. Our Contribution

In this paper, our primary contribution lies in the analysis of the scheme proposed by Sanyashi et al. [4], specifically to look at the COA security of the encryption scheme. We prove that the encryption scheme is *not* COA secure. It is argued by the authors that the product of a vector, which comprises of affine-shifted data and random nonces, when multiplied with

a random secret matrix would yield indistinguishable random ciphertexts. In this work, we revisit the validity of this analysis from the point of view of ciphertexts reflecting differences in the underlying plaintexts, and, hence, find that the assumption is not valid.

We present emperical evidence to suggest that the scheme proposed by Sanyashi et al. [4] does not provide COA security. We implement a COA attack on this scheme, demonstrating that the attacker's distinguishing advantage is consistently ≈ 1, in all trials of a few hundred test runs. This finding substantiates our argument concerning the absence of COA security for the encryption scheme in [4]. The experimental results are described in Section IV. Our code is available at https://github.com/Santosh-Upadhyaya/ICCN-INFOCOM-24/blob/main/coa-attack.ipynb

B. Organization of the Paper

In Section II, we present a recap of the protocol by Sanyashi et al. [4]. Section III presents the details of the COA indistinguishability game and our COA attack. In Section IV, we present the details and results of our experiment. Section V concludes the paper.

II. RECAP OF THE PROTOCOL BY SANYASHI ET AL.

The scheme put forth by Sanyashi et al. [4] serves as an improvement over the one proposed by Zhu et al. [3]. In the subsequent sections, we will summarize the Key Generation, Encryption, and Decryption components of the scheme from Sanyashi et al. [4]. The aspects of Query Encryption and Secure k-NN computation are not included in this recap as they do not pertain directly to the current study.

A. Key generation

Consider a database Δ that comprises of n vectors with d dimensions, $n, d \in \mathbb{Z}^+$, the set of positive integers. The elements of the database are assumed to be real numbers.
Remark: Throughout this work, we consider real numbers to be sampled uniform randomly and independently from a finite set with suitable bounds, and are represented in a fixed-point scaled integer representation in the underlying hardware architecture.
The key generation phase has the following steps.
- The public parameters c and ϵ are generated, where $c, \epsilon \in \mathbb{Z}^+$.
- A secret vector $\mathbf{s} \in \mathbb{R}^{d+1}$ is uniform randomly sampled.
- A secret matrix $\mathbf{M} \in \mathbb{R}^{\eta \times \eta}$ is uniform randomly sampled.
- A secret vector $\mathbf{w} \in \mathbb{R}^{\mathbf{c}}$ is uniform randomly sampled.

Finally, $(\mathbf{s}, \mathbf{M}, \mathbf{w})$ is the secret key for the encryption scheme.

B. Encryption

During data encryption, DO generates a nonce vector $\mathbf{z}$ of length ϵ. Consider the data to be encrypted be $\mathbf{m}_i$ such that $\mathbf{m}_i = (m_{i,1}, m_{i,2}, \cdots, m_{i,d}) \in \mathbb{R}^d$. Let $\mathbf{w}$ be a (fixed) secret vector of length c, and let $\mathbf{s}$ be a (fixed) secret vector of length $d + 1$ such that the elements of $\mathbf{s}$ are $s_1, s_2, \cdots, s_{d+1}$. The

data is pre-processed (affine shifted by $\mathbf{s}$) and subjected to encryption by vector multiplication of the pre-processed data with the inverse of a (fixed) secret matrix $\mathbf{M} \in \mathbb{R}^{\eta \times \eta}$, to get a ciphertext $\mathbf{c}_i$ of length η, where

$$
\mathbf{c}_i = (s_1 - 2m_{i,1}, \cdots, s_d - 2m_{i,d}, s_{d+1} + \\
||m_i||^2, \mathbf{w}, \mathbf{z}_i)) \times \mathbf{M}^{-1} \quad (1)
$$

The size of each ciphertext vector is $\eta = d + 1 + c + \epsilon$.

C. Decryption

During the decryption process, the nonce vector is first recovered by computing

$$
\mathbf{m}'_i = (\mathbf{c}_i \times \mathbf{M}),
$$

where $\mathbf{m}'_i = (m''_i, s_{d+1} + ||m_i||^2, \mathbf{w}, \mathbf{z}_i)$, and $\mathbf{m}''_i = (s_1 - 2m_{i,1}, \cdots, s_d - 2m_{i,d})$. The individual data elements are recovered as

$$
m_{i,j} = \frac{s_j - m''_{i,j}}{2}. \quad (2)
$$

III. COA ATTACK ON THE SCHEME OF SANYASHI ET AL.

We consider a semi-honest adversarial model in our attack. The adversary follows the protocol as in the COA indistinguishability game but tries to glean more information than is available per the protocol. In this section, we give a quick recap of the COA indistinguishability game, followed by our attack on [4].

A. Recap of COA Indistinguishability Game

The COA indistinguishability game [9], illustrated in Fig. 1, involves the following steps:
- Firstly, the adversary submits two multi-messages (sets of messages), denoted as $\mathbf{a}$ and $\mathbf{b}$, to the verifier.
- The verifier then randomly selects $b \in \{0, 1\}$.
- Next, the verifier generates the key utilizing the key generation method outlined in Section II-A. The verifier encrypts $\mathbf{a}$ if $b = 0$, otherwise encrypts $\mathbf{b}$, using the encryption procedure detailed in Section II-B.
- The ciphertexts are then sent to the adversary.
- The adversary applies its resources and outputs b', which is subsequently returned to the verifier.

The encryption scheme is COA secure, if the probability of $(b' = b) \leq \frac{1}{2} + negl()$, where $negl()$ is a negligible function of the security parameter [18].

B. Our COA Attack

The main idea behind our attack is as follows. Since the matrix-vector multiplication is a linear function, the difference of ciphertexts results in an approximate encryption of the difference of the underlying plaintexts with the secrets $\mathbf{s}$ and $\mathbf{w}$ now being canceled. Note that if the two message vectors are identical, then the difference of such ciphertexts results in an approximate encryption of the $\mathbf{0}$ vector with the ciphertext being the difference of randomly chosen vectors
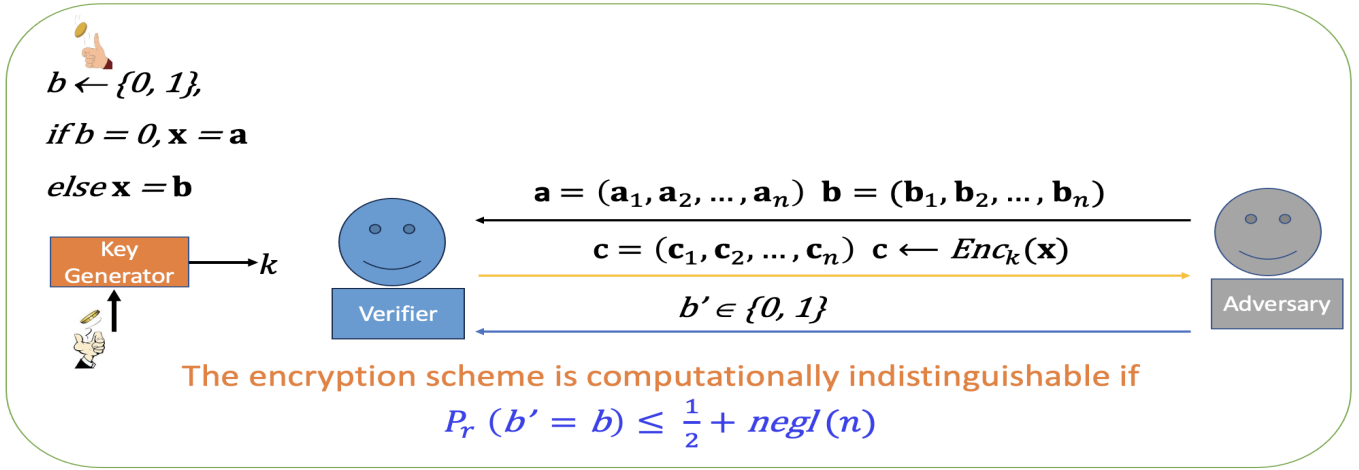
Fig. 1. COA Indistinguishibility Game

from the subspace spanned by the last ϵ columns of the secret matrix $\mathbf{M}$. Note that if the distinguisher sees many (ciphertext) vectors from the subspace, then it can readily recover the basis of the subspace. Using this information, it can then readily determine if a new difference ciphertext belongs to the subspace, and hence, whether it is an approximate ciphertext of the $\mathbf{0}$ vector. On the contrary, if the underlying message vectors are distinct, then the resulting ciphertext will not be from the above subspace. This is the basis of our COA attack.

Consider two multi-messages $\mathbf{a} = (\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_n)$ and $\mathbf{b} = (\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n)$, $n > \eta$ and each $\mathbf{a}_i, \mathbf{b}_i \in \mathbb{R}^d$. Encryption of $\mathbf{a}$ (or $\mathbf{b}$) results in a set $\mathbf{c_a}$ (or $\mathbf{c_b}$) of ciphertexts. The adversary has access to one randomly picked set of ciphertexts $\mathbf{c} \in \{\mathbf{c_a}, \mathbf{c_b}\}$ and let $\mathbf{c} = (\mathbf{c_1}, \mathbf{c_2}, \cdots, \mathbf{c_n})$. We note here that each individual message encryption, namely $\mathbf{c}_i$, is a vector of size η. Just as required in the COA indistinguishability game, the adversary in our attack is given access to one of two sets of ciphertexts, and the semantic security of the underlying cryptosystem is determined by whether the adversary can correctly determine to which of the two multi-messages the set of ciphertexts correspond to.

As per the indistinguishability game, the adversary picks two multi-messages $\mathbf{a}$ and $\mathbf{b}$ in the following way and sends them to the verifier. Let each individual message, $\mathbf{a}_i \in \mathbf{a}$, be a $\mathbf{0}$ vector of size d, and each $\mathbf{b}_i \in \mathbf{b}$ be a vector of size d with elements drawn randomly from $\mathbb{R}$, such that every $\mathbf{b}_i$ is *distinct* from one another. As mentioned before, after encryption, the adversary is given only one set of ciphertexts (either the encryption of $\mathbf{a}$ or that of $\mathbf{b}$, randomly picked and unknown to the adversary).

Computing Differences of Ciphertexts: We know that each $\mathbf{c}_i \in \mathbb{R}^\eta$, is an encryption of a message in $\mathbb{R}^d$. We pick η number of pairs $(\mathbf{c}_i, \mathbf{c}_j)$, $(i > j)$, from the total of $^nC_2$ possible pairs with $n > \eta$. Consider the μth pair $(\mathbf{c}_i, \mathbf{c}_j)$, $1 \leq \mu \leq \eta$, in some random ordering, and let $\boldsymbol{\delta}_\mu = (\mathbf{c}_i - \mathbf{c}_j)$, the difference of ciphertexts. Similarly, we compute $\boldsymbol{\delta}_1$ to $\boldsymbol{\delta}_\eta$.

W.l.o.g consider one such difference $\boldsymbol{\delta}_\mu = (\mathbf{c}_i - \mathbf{c}_j)$. Let $\mathbf{c}_i, \mathbf{c}_j$ be the encryptions of messages $\mathbf{m}_i, \mathbf{m}_j \in \mathbb{R}^d$, respectively. In other words, $\mathbf{c}_i$ is the encryption of $(m_{i,1}, m_{i,2}, \cdots, m_{i,d})$ and $\mathbf{c}_j$ is the encryption of $(m_{j,1}, m_{j,2}, \cdots, m_{j,d})$. Then, from Eqn. (1), we see that

$$\mathbf{c}_i = (s_1 - 2m_{i,1}, \cdots, s_d - 2m_{i,d}, s_{d+1} + \\ ||m_i||^2, \mathbf{w}, \mathbf{z}_i) \times \mathbf{M}^{-1}, \quad (3)$$

$$\mathbf{c}_j = (s_1 - 2m_{j,1}, \cdots, s_d - 2m_{j,d}, s_{d+1} + \\ ||m_j||^2, \mathbf{w}, \mathbf{z}_j) \times \mathbf{M}^{-1}, \quad (4)$$

therefore,

$$\boldsymbol{\delta}_\mu = (-2m_{i,1} + 2m_{j,1}, \cdots, -2m_{i,d} + 2m_{j,d}, \\ ||m_i||^2 - ||m_j||^2, \mathbf{0}, \mathbf{z}_i - \mathbf{z}_j) \times \mathbf{M}^{-1}. \quad (5)$$

In other words, in each $\boldsymbol{\delta}_i$ the secrets $\mathbf{s}$ and $\mathbf{w}$ get canceled, resulting in the form given in Eqn. 5.

In the following, we use $\boldsymbol{\delta}_i$ to mean a particular instance of the encryption difference, with $1 \leq i \leq \eta$. Recall that $\boldsymbol{\delta}_i \in \mathbb{R}^\eta$, $\eta = d + 1 + c + \epsilon$, $c = |\mathbf{w}|$ and $\epsilon = |\mathbf{z}|$ and $\boldsymbol{\delta_i}$ can be written (from Eqn. 5) as

$$\boldsymbol{\delta}_i = (r_1, r_2, \cdots, r_d, r_{d+1}, \mathbf{0}, \mathbf{z'}) \times \mathbf{M}^{-1} \quad (6)$$

In the case that encryption of $\mathbf{a}$ was provided to the adversary, then $r_i = 0$, $1 \leq i \leq d + 1$, whereas if the encryption of $\mathbf{b}$ was provided to the adversary, then $r_i \in \mathbb{R}, 1 \leq i \leq d + 1$ are expected to be randomly distributed. This is because the corresponding message vectors were randomly chosen, and, hence, their difference is expected to be random.

Remark*: Just by looking at the $\boldsymbol{\delta}_i$ vectors in Eqn. 6, the underlying r_i values remain unknown to the adversary as they are masked by the secret matrix $\mathbf{M}^{-1}$ and $\mathbf{z'}$.*

Description of our attack: We pick ϵ number of $\boldsymbol{\delta}_i$s, in no particular order, and form a set $\boldsymbol{\beta} = (\boldsymbol{\delta}_1, \cdots, \boldsymbol{\delta}_\epsilon)$. We initialize

two counters in_span and not_in_span to 0. Then, we start with $\boldsymbol{\delta}_{\epsilon+1}$ and check if it is in the span of $\boldsymbol{\beta}$. If not, then we add $\boldsymbol{\delta}_{\epsilon+1}$ to $\boldsymbol{\beta}$ and increment not_in_span counter. Otherwise, we increment in_span counter. We repeat this process for $\boldsymbol{\delta}_{\epsilon+2}$ to $\boldsymbol{\delta}_{3\epsilon}$, a total of $2\epsilon - 1$ times. Finally, if $in\_span > \epsilon$, our algorithm returns 0, else it returns 1.

Analysis of our attack: If the encryption of $\mathbf{a}$ was provided to the adversary, and since each of the r_i in Eqn. (6) are 0s, the first $d+1+c$ elements of $\boldsymbol{\delta}_i$ would just be a linear combination of $\mathbf{z}'$ combined with the last ϵ columns of $\mathbf{M}^{-1}$. When we consider ϵ number of linearly independent $\boldsymbol{\delta}_i$, they would constitute a basis for the vector space of $\mathbf{z}'$. Since we have η number of $\boldsymbol{\delta}_i$ and $\eta >> \epsilon$, we will be able to find ϵ number of linearly independent $\boldsymbol{\delta}_i$ vectors with a high probability. To simplify this check, we start with the set $\boldsymbol{\beta}$ as described above, and then we verify if every new $\boldsymbol{\delta}_i$ vector that we pick is in the span of $\boldsymbol{\beta}$, which would be the case if $\boldsymbol{\beta}$ had only linearly independent vectors. Otherwise, we add this vector to $\boldsymbol{\beta}$ and after ϵ many iterations, we are guaranteed to have at least ϵ linearly independent vectors in $\boldsymbol{\beta}$ and all subsequent ϵ many $\boldsymbol{\delta}_i$ vectors would be in the span of $\boldsymbol{\beta}$ resulting in $in\_span \geq \epsilon$.

On the other hand, if the adversary were given the encryption of $\mathbf{b}$, then the first d elements of each $\boldsymbol{\delta}_i$ would be random. Therefore, with a high probability each of the $\boldsymbol{\delta}_i$ vectors are expected to be linearly independent of the vectors of $\boldsymbol{\beta}$, even if $|\boldsymbol{\beta}| = 2\epsilon$, and so they will not be present in the span of $\boldsymbol{\beta}$. Hence, with a high probability, we would have $in\_span < \epsilon$ and $not\_in\_span \geq \epsilon$. Our attack method is given in Algorithm 1.

Given the fact that a random matrix is used to multiply a vector consisting of affine-shifted data and nonces, it is natural to expect that the resulting vectors (ciphertexts in this case) are sufficiently randomized. Similarly, multiplication by a random matrix was expected to provide randomness even when differences in ciphertexts are obtained, thus rendering them indistinguishable from random. However, our method above shows that this is indeed *not* the case.

IV. RESULTS

The proof-of-concept code for the attack is written in SageMath [19] to simulate the COA indistinguishability game. Our code consists of Key Generation, Encryption and CoA attack algorithms as outlined in Sections II and III-B. An auxiliary wrapper function reads the iteration count which is a user-defined value to mean the number of trials of the experiment to be repeated. In the experiment, values d, c, ϵ are considered as configurable global parameters and $\eta = d+1+c+\epsilon$. In each iteration, a choice bit, $\mathbf{0}$ or $\mathbf{1}$, is picked uniformly random. If $\mathbf{0}$ is chosen, a multi-message of size $(\eta + 1)$ consisting of vectors of all $\mathbf{0}$s is used for encryption, otherwise, a multi-message of size $(\eta+1)$ with distinct vectors with random elements is used for encryption. This part mimics the role of the verifier and outputs the ciphertext.

The attack algorithm obtains the ciphertext from the verifier and executes the method described in Sec. III-B. It finally outputs a bit as the result which is returned to the verifier.

Algorithm 1 Attack on Encrypted Data

Require: $\mathbf{c}_1, \cdots, \mathbf{c}_n$ {Ciphertext of a multi-message of length n $(n > \eta >> \epsilon)$}
1: Initialize $not\_in\_span\_cnt \leftarrow 0$, $in\_span\_cnt \leftarrow 0$
2: **for** $i = 0$ to $n - 1$ **do**
3:     $\delta_i \leftarrow \mathbf{c}_{i+1} - \mathbf{c}_i$ {Compute difference of ciphertexts}
4: **end for**
5: $\beta = \{\}$
6: **for** $i = 0$ to $\epsilon - 1$ **do**
7:     ADD δ_i to set β {Pick ϵ number of differences}
8: **end for**
9: **for** $i = \epsilon$ to $3\epsilon - 1$ **do**
10:     **if** δ_i in LINEAR_SPAN(β) **then**
11:         $in\_span\_cnt \leftarrow in\_span\_cnt + 1$
12:     **else**
13:         $not\_in\_span\_cnt \leftarrow not\_in\_span\_cnt + 1$
14:         ADD δ_i to set β
15:     **end if**
16: **end for**
17: **if** $in\_span\_cnt \geq \epsilon$ **then**
18:     **return** 0
19: **else**
20:     **return** 1
21: **end if**

Thorough testing of the code was done with various values of d and we iterated the experiment 512 times. Our goal was to validate the accuracy of our attack algorithm. This testing involved setting $c = 5$ and $\epsilon = 5$. The attack algorithm successfully distinguished the ciphertexts with 100% accuracy. As stated in Sec. 2.2 of [3], the value of d is expected to be 100, and our testing was conducted with d values ranging from 8 to 128. As given in Sec. 7.1 of [4], the security parameters c and ϵ can be set to 5, and our experiments with various d values were conducted with $c = \epsilon = 5$. The average execution time for a single attack is provided in Table I.

TABLE I
PERFORMANCE OF ATTACK ALGORITHM

Value of d	Average Execution time of attack (sec)
8	0.08
16	0.09
32	0.26
64	0.94
128	3.83

V. CONCLUSION

The primary focus of this research was to devise a COA attack on the scheme used in [4], based on the multi-message COA indistinguishability game. The proposed attack revealed that the attacker's distinguishing advantage is ≈ 1. This invalidates the assertions made in Lemma 2 in [4]. This investigation highlights the necessity for comprehensive security assessments in the design of cryptographic systems. It would be

interesting to explore modifications to the encryption scheme of [4] that would thwart our attack. More importantly, such a proposal should be accompanied with a rigorous security analysis based only on the hardness assumptions of well-established problems.

REFERENCES

[1] Ahmad, A., et al.: Parallel query execution over encrypted data in database-as-a-service (DaaS). J. Supercomput. 75, 2269–2288 (2019)

[2] Wong, Wai Kit and Cheung, David Wai-lok and Kao, Ben and Mamoulis, Nikos on Secure KNN Computation on Encrypted Databases, 2009, 9781605585512, Association for Computing Machinery, Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, pp.139–152, https://doi.org/10.1145/1559845.1559862,

[3] Zhu, Y., Huang, Z., Takagi, T.: Secure and controllable k-NN query over encrypted cloud data with key confidentiality. J. Parallel Distrib. Comput. 89, 1–12 (2016)

[4] Sanyashi, T., Boran, N.K., Singh, V. (2023). Secure KNN Computation on Cloud. In: Muthukkumarasamy, V., Sudarsan, S.D., Shyamasundar, R.K. (eds) Information Systems Security. ICISS 2023. Lecture Notes in Computer Science, vol 14424. Springer, Cham. https://doi.org/10.1007/978-3-031-49099-6-12

[5] Abbas Acar, Hidayet Aksu, A. Selcuk Uluagac, and Mauro Conti. 2018. A Survey on Homomorphic Encryption Schemes: Theory and Implementation. ACM Comput. Surv. 51, 4, Article 79 (July 2019), 35 pages. https://doi.org/10.1145/3214303

[6] Z. H. Mahmood and M. K. Ibrahem, "New Fully Homomorphic Encryption Scheme Based on Multistage Partial Homomorphic Encryption Applied in Cloud Computing," 2018 1st Annual International Conference on Information and Sciences (AiCIS), Fallujah, Iraq, 2018, pp. 182-186, doi: 10.1109/AiCIS.2018.00043.

[7] R. Sendhil and A. Amuthan, "A Descriptive Study on Homomorphic Encryption Schemes for Enhancing Security in Fog Computing," 2020 International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2020, pp.738-743, doi:10.1109/ICOSEC49089.2020.9215422

[8] Golightly L, Chang V, Xu QA, Gao X, Liu BS. Adoption of cloud computing as innovation in the organization. International Journal of Engineering Business Management , 2022, doi:10.1177/18479790221093992

[9] Jonathan Katz, Yehuda Lindell, Introduction to Modern Cryptography: Second Edition, Principles and Protocols (ChapmanHall/CRC Cryptography and Network Security Series)", 2007

[10] D. S. Linthicum, "Connecting Fog and Cloud Computing," in IEEE Cloud Computing, vol. 4, no. 2, pp. 18-20, March-April 2017, doi: 10.1109/MCC.2017.37

[11] N. Zhou, F. Dufour, V. Bode, P. Zinterhof, N. J. Hammer and D. Kranzlmüller, "Towards Confidential Computing: A Secure Cloud Architecture for Big Data Analytics and AI," 2023 IEEE 16th International Conference on Cloud Computing (CLOUD), Chicago, IL, USA, 2023, pp. 293-295, doi: 10.1109/CLOUD60044.2023.00042.

[12] R. Li, A. X. Liu, Y. Liu, H. Xu and H. Yuan, "Insecurity and Hardness of Nearest Neighbor Queries Over Encrypted Data," 2019 IEEE 35th International Conference on Data Engineering (ICDE), Macao, China, 2019, pp. 1614-1617, doi: 10.1109/ICDE.2019.00155.

[13] Sanyashi Tikaram and Menezes Bernard. (2023)."Secure Computation over Encrypted Databases".

[14] H. -J. Kim, H. -J. Lee and J. -W. Chang, "A Secure and Efficient Query Processing Algorithm Over Encrypted Database in Cloud Computing," 2021 IEEE International Conference on Big Data and Smart Computing (BigComp), Jeju Island, Korea (South), 2021, pp. 219-225, doi: 10.1109/BigComp51126.2021.00049.

[15] S. Almakdi and B. Panda, "A Secure Model to Execute Queries Over Encrypted Databases in the Cloud," 2019 IEEE International Conference on Smart Cloud (SmartCloud), Tokyo, Japan, 2019, pp. 31-36, doi: 10.1109/SmartCloud.2019.00015.

[16] M. Kesarwani, A. Kaul, S. Braghin, N. Holohan and S. Antonatos, "Secure k-Anonymization over Encrypted Databases," 2021 IEEE 14th International Conference on Cloud Computing (CLOUD), Chicago, IL, USA, 2021, pp. 20-30, doi: 10.1109/CLOUD53861.2021.00015.

[17] Chunsheng, Gu and Jixing, Gu, "Known-plaintext attack on secure kNN computation on encrypted databases" 2014, Security and Communication Networks.

[18] https://en.wikipedia.org/wiki/Negligible_function

[19] https://www.sagemath.org