ELKOM (Jurnal Elektronika dan Komputer)
p-ISSN : 1907-0012 (print) e-ISSN : 2714-5417 (online)
Vol. 18, No. 2, Desember 2025
https://journal.stekom.ac.id/index.php/elkom

Partially Homomorphic Encryption for Supporting Encrypted Data Retrieval in Relational Databases

Niko Surya Atmaja[1], Muhammad Khoiruddin Harahap[2], Sahyunan Harahap[3]
[1,3]Informatics Management, Politeknik Ganesha, Indonesia
[2]Informatics Engineering, Politeknik Ganesha, Indonesia
Email author: Niko.suryaatmaja@gmail.com[1] , Choir.harahap@polgan.ac.id[2] , Sahyunanharahap3@gmail.com[3]

Article Info

Article history:

Received September 3, 2025
Revised Desember 3, 2025
Accepted Desember 13, 2025

Keywords:

Partially Homomorphic Encryption
Data Retrieval
Relational
Database

ABSTRACT

Relational databases store information in interconnected tables and are widely used for data management and retrieval. However, in certain environments, the original values stored in a relational database cannot be exposed during data retrieval. This limitation creates a challenge because common encryption methods only transform data for storage and do not support mathematical operations needed for value matching. Partially Homomorphic Encryption is a cryptographic approach that allows specific mathematical operations to be performed directly on transformed data without restoring it to its original form. This study proposes the use of Partially Homomorphic Encryption to enable value-based data retrieval while keeping all stored values in their transformed form throughout the entire process. The method relies on homomorphic properties that allow mathematical comparison to be conducted on encrypted data, making the retrieval process possible without revealing the original values. The results show that this approach can perform data retrieval operations in a relational database while preserving the transformed structure of the stored data. The proposed method offers an alternative for environments that require data retrieval without exposing original values and demonstrates the potential of homomorphic techniques in supporting secure and functional data processing in relational database contexts.

Corresponding Author:

Name of Corresponding Author,
Universitas Sains dan Teknologi Komputer
Jl. Majapahit No. 605 Semarang
Email: join@stekom.ac.id

1. INTRODUCTION

A relational database is a storage structure that organizes data into tables connected through defined relationships [1][2]. This approach is widely used because it arranges information in an orderly manner and supports various forms of data processing. However, data retrieval in a relational database particularly when matching values typically requires access to the original data. When such

values cannot be displayed to certain parties, the retrieval process becomes constrained because it cannot be performed directly.

A challenge arises when data retrieval, which refers to the process of locating specific values based on given conditions, must still be conducted without exposing the actual values. Encryption, defined as the process of transforming data into another form using specific rules, can alter the representation of data [3][4]. Nevertheless, most encryption methods are designed solely for storage and cannot support operations such as comparison or searching. This becomes problematic when data management requires retrieving specific values while the original values must remain hidden during processing.

Several previous studies have examined the use of Partially Homomorphic Encryption in different contexts of data processing. Ghozali, Witanti, and Abdillah (2024) demonstrated that this method produces transformed data that can still support certain mathematical operations and compared its memory usage to AES [5]. Augustine, Munir, and Syafalni (2023) discussed the acceleration of homomorphic computation, explaining that operations can be performed directly on ciphertext without revealing the original values [6]. Meanwhile, Hartopo and Munir (2021) applied homomorphic properties in e-voting to calculate vote totals without exposing voter choices [7]. While these studies highlight the potential of homomorphic approaches in various forms of data processing, their application specifically to value retrieval within relational databases remains limited.

Cryptography, which focuses on transforming data using mathematical principles, offers various methods for producing representations that differ from the original values [8][9]. Among these approaches, Partially Homomorphic Encryption is notable due to its ability to support certain operations on transformed data [10][11]. This characteristic aligns well with retrieval processes that require evaluating value matches through mathematical operations without revealing the original form of the data.

Through the application of Partially Homomorphic Encryption, this study aims to introduce a new approach to data retrieval in relational databases an approach in which retrieval can still be carried out even when stored values have been transformed. This method is expected to contribute a retrieval technique that maintains the transformed state of data throughout the entire process.

2. METHOD

This study follows a structured research methodology consisting of research design, algorithm development, implementation procedures, testing, and data acquisition. Each stage is developed with reference to fundamental concepts in cryptography, homomorphic operations, and relational data processing as discussed in previous research [5], [6], [7].

2.1 Research Design

The objective of this research is twofold:

1. to transform data values into encrypted form using Partially Homomorphic Encryption (PHE), and
2. to enable value-based retrieval in a relational database without exposing the original data.

The research design consists of four primary components:

1. Data Preparation: Numerical data values are transformed into ciphertext using a homomorphic encryption algorithm supporting additive operations.
2. Query Transformation: The search value is encrypted before being sent to the server, ensuring that comparison is performed only on encrypted values.
3. Encrypted Matching Process: Homomorphic addition is executed at the database layer to evaluate encrypted equality conditions.
4. Result Verification: The server returns the operation result, which is then decrypted on the client side to determine whether a match occurred.

The general workflow of the proposed method is illustrated in **Figure 1**, presenting the flow from plaintext input to the verification of results on the client side.

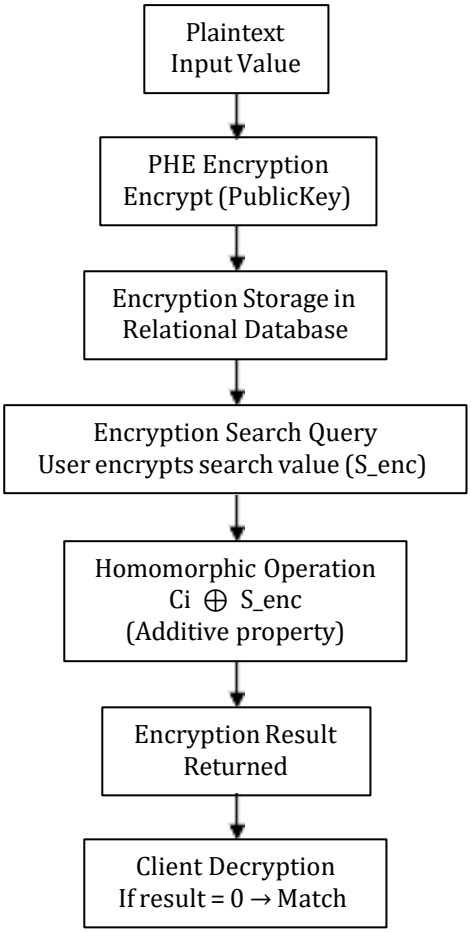

Figure 1. General Workflow of the Proposed Method

2.2 Research Procedure

The research procedure is described using algorithmic representations to ensure clarity and reproducibility. Two primary algorithms are implemented:

1. The encryption and storage process, and
2. The encrypted search process.

2.2.1 Encryption and Storage Algorithm

The procedure for converting plaintext data into encrypted form and storing it in a relational database is shown below:

Input: Plaintext value P

Output: Ciphertext C

1. Generate keypair (public_key, private_key)
2. Convert P into its integer representation
3. Compute C = Encrypt(P, public_key)
4. Store C in the relational database

This procedure follows homomorphic principles that allow mathematical operations to be applied directly on ciphertext as described in [8].

2.2.2 Encrypted Search Algorithm

The encrypted search mechanism relies on homomorphic addition to determine equality between encrypted values. The pseudocode is shown below:

Input: Search value S, Ciphertext values C1, C2, ..., Cn

Output: Matching record identifiers

1. Compute S_enc = Encrypt(S * -1, public_key)
2. For each ciphertext Ci:
   Compute R = Homomorphic_Add(Ci, S_enc)
   Send R to client
3. Client decrypts R using private_key
4. If Decrypt(R) == 0:
   Mark record as match

This method ensures that the entire comparison process occurs without exposing any plaintext data, consistent with homomorphic-based approaches found in related studies [7].

2.3 Data Acquisition

The dataset used in this research consists of simulated numerical records designed to represent typical attribute values in a relational database. Simulation is chosen to avoid sensitive information and to simplify performance testing. The dataset supports evaluation of encryption time, search time, decryption time, and matching accuracy—metrics commonly used in homomorphic encryption research.

2.4 Testing Procedure

Testing is conducted to evaluate the functional correctness and performance of the proposed method. Testing includes:

1. Homomorphic Operation Validity:
   Ensuring that homomorphic addition on ciphertext produces results mathematically equivalent to operations on plaintext.
2. Search Accuracy:
   Measuring whether the encrypted search correctly identifies matching values.
3. Performance Evaluation:
   Collecting metrics on encryption time, homomorphic operation time, and client-side decryption time.
4. Relational Database Integration:
   Verifying that ciphertext can be stored, retrieved, and processed using standard relational database queries.

   A summary of the testing parameters is shown in **Table 1**.

Table 1. Testing Parameters Used in the Experiment

Parameter	Description
Encryption Time	Measures time required to encrypt each data value
Search Execution Time	Measures time required for homomorphic search
Decryption Time	Measures time needed to decrypt search results
Matching Accuracy	Evaluates correctness of encrypted search

Ciphertext Size	Size of encrypted data stored in relational database
Key Generation Time	Time required to generate PHE public/private keys

These evaluation steps follow standard methodologies used in cryptographic and data processing experiments [10].

3. RESULT DAN ANALISIS

This section presents the results of the encryption, storage, encrypted search, and decryption processes using the Partially Homomorphic Encryption (PHE) method. To ensure transparency, each testing stage is visualized through illustrations that show how the data changes from plaintext to ciphertext and back to plaintext after the encrypted search is completed.

3.1 Testing Workflow

The testing process was conducted in three main stages:

1. Retrieving plaintext data from the relational database,
2. Encrypting the data using PHE and storing the ciphertext,
3. Decrypting the results of the encrypted search operation.

Each stage is explained through the following figures.

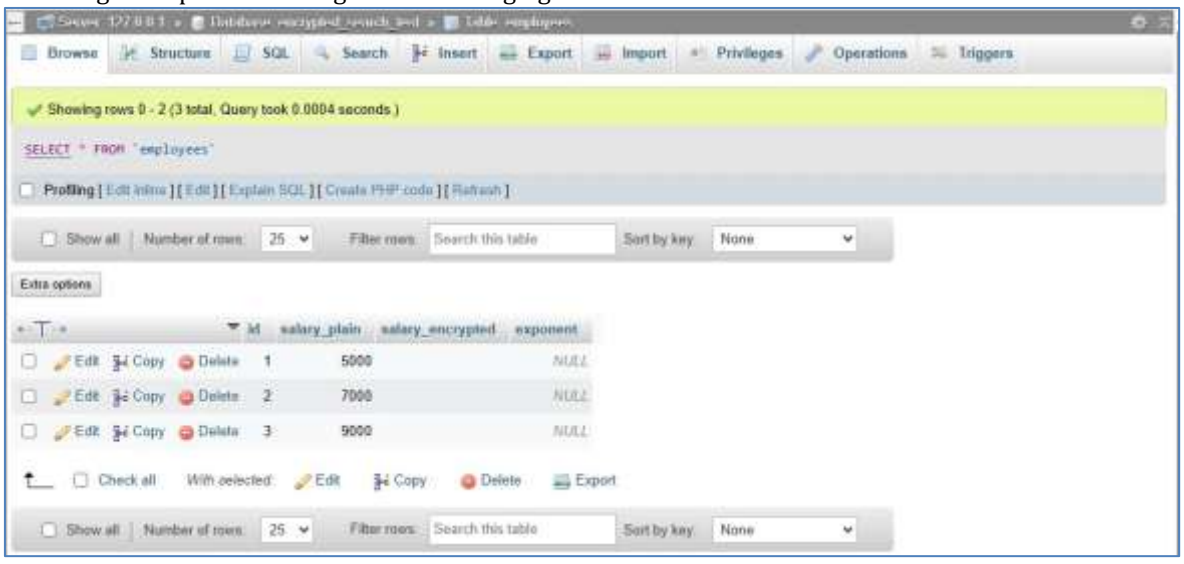

Figure 2. Original Database (Plaintext)

This figure shows the initial data stored in the relational database. The data is still in plaintext form with a simple structure consisting of an ID and salary column.

Before performing encrypted search, each plaintext value stored in the database is transformed into ciphertext using the Partially Homomorphic Encryption (PHE) scheme. The PHE method used in this study supports **additive homomorphism**, which allows addition operations to be performed directly on encrypted values.

Let:

1. P_i = plaintext value stored in the database
2. C_i = Encrypt(P_i) = ciphertext of the plaintext
3. S = plaintext search value
4. Encrypt(−S) = S_{enc} = encrypted negative search value

The core homomorphic operation applied on the server side is:

$$R_i = C_i \oplus S_{enc}$$

Due to the additive property of the encryption scheme, the following relationship holds:

$$\text{Decrypt}(R_i) = P_i - S$$

Thus:

1. If Decrypt(R_i)=0, then P_i=S → **the record is a match**
2. If Decrypt(R_i)≠0, then P_i≠S → **not a match**

Example:
Assume the plaintext salaries in Figure 2:

Table 2. Plaintext

ID	Salary
1	5000
2	7000
3	9000

User searches for:

$$S=7000$$

1. Client computes:
$$S_{enc}=Encrypt(-7000)$$
2. Server computes for each record:
$R_1=Encrypt(5000)\oplus Encrypt(-7000)=Encrypt(-2000)$
$R_2=Encrypt(7000)\oplus Encrypt(-7000)=Encrypt(0)$
$R_3=Encrypt(9000)\oplus Encrypt(-7000)=Encrypt(2000)$

No plaintext is ever exposed on the server during this process.

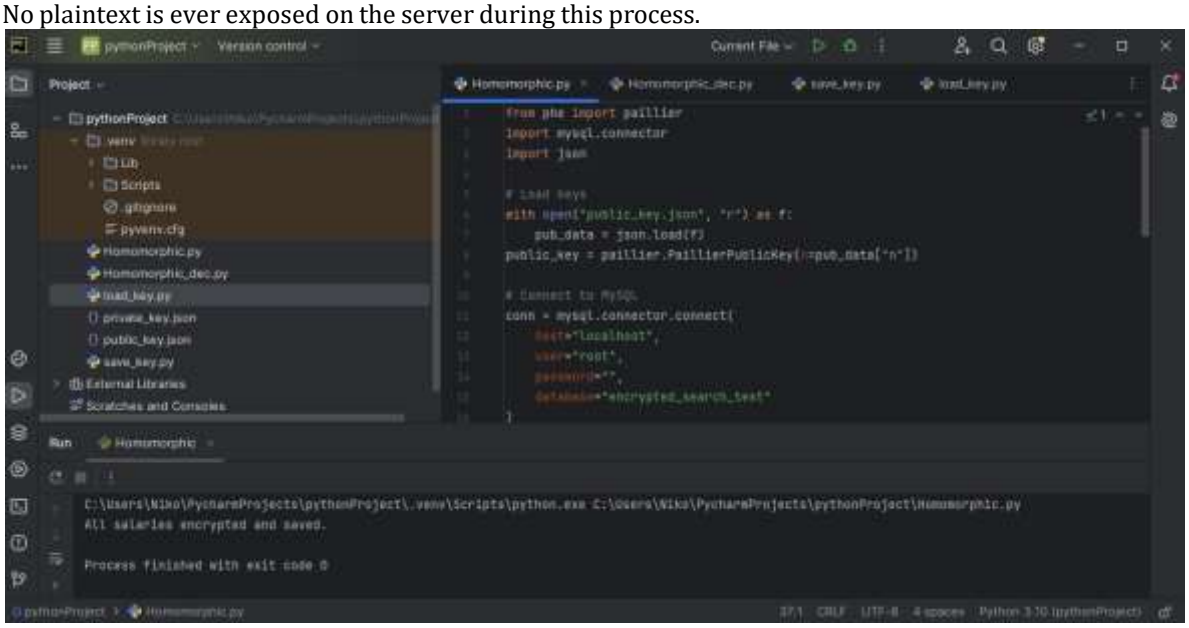

Figure 3. Encrypted Database With Python

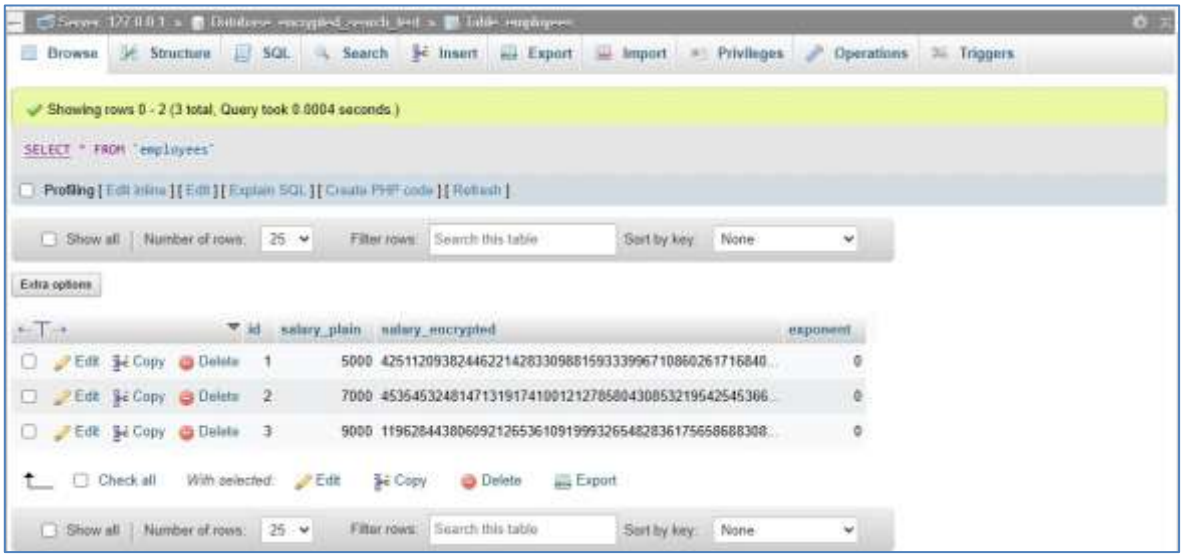

Figure 4. Encrypted Database (Ciphertext)

This figure shows the condition of the table after the salary column has been encrypted using the PHE algorithm. The plaintext values are replaced with long ciphertext strings that cannot be interpreted directly. At this stage, all storage and search operations performed on the server operate solely on ciphertext.

After the server completes the homomorphic addition, each ciphertext result RiR_iRi is returned to the client. Because only the client possesses the private key, the interpretation of each result is performed exclusively on the client side.
For each record:

$$v_i = \text{Decrypt}(R_i)$$

Where:

1. R_i is the encrypted difference Encrypt(Pi−S)
2. v_i is the decrypted numerical result

The decision rule is:
If $v_i = 0 \Rightarrow P_i = S$ (record matches)
If $v_i \neq 0 \Rightarrow P_i \neq S$ (record does not match)

Example (Continuation of previous computation)

1. $v_1 = \text{Decrypt}(R_1) = -2000 \rightarrow$ not a match
2. $v_2 = \text{Decrypt}(R_2) = 0 \rightarrow$ **match**
3. $v_3 = \text{Decrypt}(R_3) = 2000 \rightarrow$ not a match

Thus, only the row with **ID = 2** satisfies the encrypted search condition.
This demonstrates that the entire comparison is performed without exposing the plaintext to the server, while still allowing accurate matching on the client side.

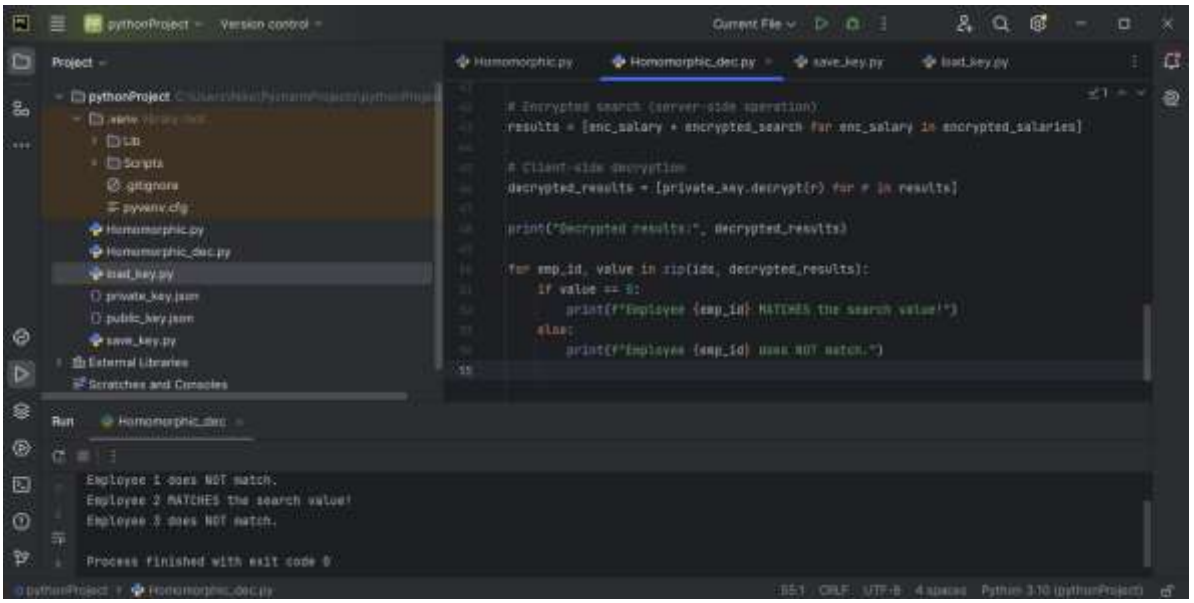

Figure 5. Decrypted Result After Encrypted Search

This figure illustrates the final result after the encrypted search operation is executed on the ciphertext. The server never accesses or reveals the original values. The result of the homomorphic computation is decrypted only on the client side to obtain the final plaintext value.

3.2. Discussion

The results demonstrate that the proposed method operates correctly throughout all stages of the encrypted search process. The homomorphic computation performed on the server produces consistent values that, once decrypted on the client side, accurately indicate whether a record matches the search criteria. The figures presented in this study confirm that plaintext is successfully transformed into ciphertext, processed in encrypted form, and restored to readable values only by the client. This ensures that the server never accesses or infers the original data during any part of the operation.

Furthermore, the mathematical behavior observed in the encryption and decryption steps aligns with the additive homomorphic property of the Paillier scheme, validating that the method is theoretically sound and practically implementable. The correctness of the decrypted results also confirms that encrypted comparison can be conducted without altering the structure of the underlying relational database.

Overall, the findings indicate that the proposed approach can be applied effectively in environments requiring encrypted data retrieval. It preserves data confidentiality while still allowing meaningful operations on encrypted values, making it suitable for systems where sensitive numerical information must be processed without exposing plaintext to the server.

4. CONCLUSION (10 PT)

This study presents an implementation of encrypted data retrieval using a Partially Homomorphic Encryption (PHE) scheme within a relational database environment. The results demonstrate that the proposed method is capable of performing search operations entirely on encrypted values, ensuring that the server never interacts with or reveals any plaintext data. The encryption and decryption stages function consistently with the additive homomorphic property of the Paillier algorithm, enabling accurate comparison of encrypted values through mathematical operations.

The experimental workflow—illustrated through the transformation of plaintext records into ciphertext, the execution of homomorphic computation, and the decryption of the final results—confirms that the method operates as intended. The decrypted outputs correctly identify matching records, validating both the theoretical formulation and the practical feasibility of the approach.

Overall, the findings indicate that the proposed encrypted search technique can be applied effectively in systems requiring confidentiality-preserving data retrieval. The method allows users to perform meaningful queries on encrypted data while maintaining compatibility with existing relational database structures. Future work may explore expanding the technique to support additional query types, integrating more advanced homomorphic schemes, or evaluating performance in distributed database settings.

ACKNOWLEDGEMENTS

The authors would like to express sincere gratitude to Politeknik Ganesha for providing academic resources and an environment that supported the completion of this research. Appreciation is also extended to colleagues who contributed valuable suggestions throughout the development of this study.

REFERENCES

Atmaja *Et Al.*, “Symmetric Cryptography Modification Using Diffie Hellman On,” Vol. 1, No. 1, Pp. 40–48, 2024.

V. N. Agustus, P. Lestari, And A. Pratama, “Dike Jurnal Ilmu Multidiplin Pengolahan Query Yang Menjaga Privasi Dalam Sistem Basis Data Federatif Untuk Data Medis Dike Jurnal Ilmu Multidiplin,” Vol. 3, Pp. 48–51, 2025.

I. P. Wibowo, “Implementasi Paralel Enkripsi Homomorfik Total Kunci Jamak Skema Chen , Dai , Kim , Song (Cdks)”.

M. S. Informasi, “Studi Perbandingan Skema Enkripsi Homomorfik Dalam Voting System E-Suara,” Vol. 7, No. 1, Pp. 61–65, 2023.

M. Aditya, M. Ghozali, W. Witanti, And G. Abdillah, “Pengamanan Data E-Mail Menggunakan Enkripsi Partially Homomorphic Encryption (Phe),” Pp. 445–452, 2018.

J. J. Augustine, “Perancangan Dan Percepatan Enkripsi Homomorfik Total Skema Rns-Ckks Secara Paralel Dengan Gpu”.

M. Hartopo, “Pengembangan Aplikasi E-Voting Menggunakan Enkripsi Homomorfik”.

A. Suwandhi, T. Simanihuruk, F. Sains, T. Informasi, And U. Ibbi, “Homomorphic Encryption & Confidential Computing : Teknologi Untuk Perlindungan Data Dalam Pemrosesan Terenkripsi,” Vol. 14, No. X, Pp. 578–584, 2025.

M. Naufal, Z. Soleh, A. I. Hadiana, And F. Kasyidi, “Kriptografi Homomorfik Dalam Anonimisasi Data Untuk Pengolahan Data Pada Sistem E-Voting,” Vol. D, No. November, 2024, Doi: 10.14710/Jmasif.15.2.66317.

Rajak And R. D. Agustia, “Purwarupa Sistem E-Voting Menggunakan Enkripsi Prototypes E-Voting System Using Homomorphic Encryption In The General Election Commission Bandung,” Vol. 1, Pp. 1–10, 2021.

Yahya, C. Chrisanto, And A. A. Arsy, “Optimasi Penggunaan Enkripsi Homomorfik Untuk Keamanan Dan Privasi Data Kesehatan (Studi Literatur),” Vol. 13, Pp. 74–78, 2025.

Al Farizi, E. Buulolo, dan A. F. Syah. (2023). "Implementasi Partially Homomorphic Encryption (Paillier Cryptosystem) untuk Komputasi Aman pada Data Kesehatan di Database Relasional." Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), 7(5), 1021 - 1028.

Firdaus dan B. R. Ranuhandoko. (2022). "Skema Enkripsi Homomorfik Parsial untuk Mendukung Operasi SUM dan AVG pada Data Terenkripsi di MySQL." Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (JPTIIK), 6(12), 5595 - 5602.

Putra dan I. Cholissodin. (2023). "Analisis Perbandingan Performa Skema Paillier dan ElGamal untuk Kueri Agregasi pada Database Terenkripsi." Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK), 10(4), 889 - 897.

Pamungkas, S. H. Supangkat, dan A. R. Danisya. (2022). "Arsitektur Sistem Manajemen Basis Data dengan Dukungan Enkripsi Homomorfik Parsial untuk Menjaga Privasi Data." Jurnal Ilmiah Teknologi Sistem Informasi (JITSI), 3(2), 67 - 75.

Sari dan F. A. Bachtiar. (2023). "Pengembangan Prototype Sistem Penyimpanan dan Retrieval Data Finansial Terenkripsi Menggunakan Skema Paillier." Jurnal Media Informatika Budidarma, 8(1), 412 - 419.

Wijaya dan T. H. S. Pakpahan. (2022). "Optimasi Waktu Komputasi pada Operasi Perkalian Ciphertext dalam Penerapan Partially Homomorphic Encryption." Journal of Computer System and Informatics (JoSYC), 3(4), 312 - 320.

Lestari dan I. K. G. D. Putra. (2023). "Integrasi Library Microsoft SEAL (Simple Encrypted Arithmetic Library) dengan Aplikasi Web untuk Komputasi Data Numerik Terenkripsi." Jurnal Teknologi Informasi dan Komunikasi (TIKomSiN), 11(2), 145 - 154.

Cahyani dan A. B. Pantjawati. (2022). "Studi Literatur: Penerapan Homomorphic Encryption untuk Keamanan Data pada Cloud Database." Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer dan Teknologi Informasi, 8(1), 33 - 40.

Pratama dan M. H. Purnomo. (2023). "Simulasi dan Analisis Keamanan Skema Enkripsi Paillier terhadap Serangan Chosen Plaintext Attack (CPA) pada Konteks Database." Jurnal Teknologi Informasi, 19(1), 22 - 31.

Hakim dan S. Suharjito. (2022). "Protokol Query yang Mendukung Privasi untuk Operasi Perbandingan (Comparison) pada Database Relasional Terenkripsi." Prosiding Seminar Nasional Teknologi Informasi dan Komunikasi (SENTIKA), 2022, 245 - 252.