

Privacy-Preserving Tax Filing System using Homomorphic Encryption for Secure Data Processing

Guggilla Badri Charan Tej¹ and Uppara Dhanunjaya²

P.G Student¹ & Assistant Professor², Computer Science and Engineering^{1&2}, Sri Krishnadevaraya University College of Engineering and Technology^{1&2}, Anantapuramu, India.

Abstract - Digital tax services handle highly sensitive personal and financial information, including taxpayer identities, income details, deductions, tax liabilities, and payment records. Although conventional encryption protects data during transmission and storage, sensitive information typically needs to be decrypted before conventional computations can be performed, potentially exposing it within the processing environment. To address this challenge, this paper proposes the design of a privacy-preserving tax filing system that employs homomorphic encryption (HE) to perform selected computations directly on encrypted taxpayer data. The proposed system consists of two primary modules: a taxpayer module and a government administration module. The taxpayer module supports registration, authentication, submission of tax-related information, tax calculation and payment, receipt generation and download, filing history review, and complaint submission. The administration module enables authorized government officials to review reports and manage complaints through a controlled interface. Paillier additive homomorphic encryption is considered for encrypted data aggregation and addition-based calculations, while operations involving multiplication, comparisons, conditional branching, and complex tax rules require alternative homomorphic encryption schemes or carefully designed hybrid processing approaches. The proposed prototype is developed using Python and the Django framework. However, the available project documentation does not include performance benchmarks or a formal security assessment. Therefore, the work focuses on the system architecture, functional design, and prototype workflow rather than claiming experimentally validated performance or comprehensive end-to-end security. The study emphasizes the importance of robust cryptographic mechanisms, secure key management, access control, and user-friendly interfaces as essential requirements for the reliable implementation of privacy-preserving digital tax administration.

Keywords: Tax filing, Homomorphic Encryption, Paillier Cryptosystem, Privacy-preserving Computation, Django, Secure Government Services.

INTRODUCTION:

Government tax portals have become essential platforms for online taxpayer registration, tax return filing, payment processing, receipt generation, and communication between taxpayers and government officials. These systems process highly confidential personal and financial information, and unauthorized disclosure may result in identity theft, financial loss, or violations of privacy. Although conventional security mechanisms, such as Transport Layer Security (TLS) and database encryption, protect information during transmission and storage, they do not fully prevent exposure when data must be decrypted for processing. For example, conventional tax calculation workflows may require the server to access plaintext values to determine taxable income, applicable deductions, and tax liabilities. Consequently, vulnerabilities in application processes, excessive administrative privileges, or improper system configurations may expose sensitive taxpayer information.

Homomorphic encryption (HE) offers an alternative approach by enabling specific computations to be performed directly on encrypted data without requiring its prior decryption. Under this model, the processing system operates on ciphertexts and produces encrypted results that can subsequently be decrypted by an authorized key holder. Paillier's public-key cryptosystem supports additive homomorphism, making it appropriate for encrypted summation and scalar multiplication operations (Paillier, 1999). In contrast, Fully Homomorphic Encryption (FHE) supports a broader range of computations, including more complex arithmetic operations, although its practical application depends on computational overhead, implementation constraints, and scheme-specific limitations (Gentry, 2009; Acar et al., 2018).

This study presents the system-level design of a privacy-preserving online tax filing portal that integrates homomorphic encryption into selected data-processing operations while maintaining conventional workflows for taxpayers and government administrators. The proposed architecture identifies opportunities to perform computations on encrypted taxpayer information and distinguishes them from operations that require conventional security mechanisms or additional cryptographic techniques. The primary contribution of this work is a structured system design that demonstrates how homomorphic encryption can be incorporated into digital tax administration to strengthen data confidentiality while recognizing the technical, operational, and security challenges associated with practical implementation.

Objectives and Contributions:

- To design a privacy-preserving tax filing workflow that safeguards sensitive taxpayer information during selected server-side computations.
- To develop and describe the taxpayer and government administration modules, incorporating tax filing history, receipt generation, and complaint management functionalities.
- To identify tax-related computations supported by Paillier additive homomorphic encryption (HE) and distinguish them from operations requiring alternative cryptographic techniques.
- To examine security boundaries, implementation requirements, and future evaluation procedures necessary to assess the system's security, performance, and suitability for real-world deployment.

RELATED WORK:

Homomorphic encryption (HE) has evolved from early concepts of performing computations on encrypted data into a range of cryptographic techniques, including partially, somewhat, and fully homomorphic encryption. Paillier (1999) introduced a probabilistic public-key cryptosystem with an additive homomorphic property, whereby the multiplication of ciphertexts corresponds to the addition of their underlying plaintext values. This capability makes Paillier encryption suitable for encrypted summation and weighted-sum calculations. However, it does not directly support the complete evaluation of complex tax rules involving arbitrary multiplication, comparisons, or conditional operations. Gentry (2009) established a major milestone in the field by presenting the first plausible construction of fully homomorphic encryption, enabling arbitrary computations on encrypted data. Subsequent research has focused on improving computational efficiency, refining encryption schemes, and expanding their practical applicability.

Acar et al. (2018) provided a comprehensive survey of homomorphic encryption techniques, implementation challenges, and potential application domains, highlighting the importance of selecting an appropriate scheme according to computational requirements and performance constraints. Similarly, the security standard published by HomomorphicEncryption.org provides guidance on encryption schemes, security assumptions, and parameter selection, supporting informed decisions during system implementation (Albrecht et al., 2018). Collectively, these studies establish the importance of treating homomorphic encryption as a specialized privacy-enhancing mechanism that complements, rather than replaces, essential security measures such as authentication, authorization, secure software development, and conventional encryption.

Within this research, privacy-preserving computation is explored in the context of digital tax administration. The project materials describe a tax filing portal incorporating homomorphic encryption for selected data-processing operations. However, they do not provide a reproducible comparison with existing tax systems, a formally defined threat model, or quantitative performance evaluation. Accordingly, the present study focuses on the proposed system architecture, functional workflow, and potential application of homomorphic encryption, without making unsupported claims regarding measured performance improvements or experimentally verified security benefits.

Existing System and Problem Definition:

The existing tax filing workflow described in the project documentation relies on conventional encryption to safeguard taxpayer information during transmission and storage. However, when the application performs tax calculations, generates reports, or processes financial records, the relevant data must generally be decrypted before computation. This creates a potential security vulnerability because sensitive plaintext information may be exposed within the application environment, particularly in the event of application compromise, unauthorized privileged access, or improper system configuration. Therefore, protecting data at rest and in transit alone is insufficient to eliminate the risks associated with plaintext processing.

Problem Statement: How can a digital tax filing system incorporate privacy-preserving computation techniques to perform selected operations on encrypted taxpayer information, thereby reducing plaintext exposure within the server environment while maintaining essential functionalities such as tax filing, payment processing, receipt generation, filing history management, and complaint handling?

Proposed System:

The proposed system is organized into two primary modules: the taxpayer module and the government administration module. The taxpayer module provides functionalities such as account registration and authentication, profile management, submission of tax-related information, tax calculation and payment processing, receipt generation and download, access to payment and filing history, complaint submission, and secure logout. The government administration module enables authorized officials to authenticate themselves, review reports, respond to taxpayer queries, manage complaints, and securely terminate their sessions. These functional requirements are derived from the project documentation and are intended to maintain the essential operations of a digital tax filing portal.

At the cryptographic layer, taxpayer information designated for privacy-preserving computation is appropriately encoded and encrypted using a selected homomorphic encryption (HE) scheme. A computation service then performs supported operations

directly on the ciphertexts and returns encrypted results to the authorized recipient. The resulting ciphertexts can be decrypted only by the holder of the corresponding decryption key or an explicitly authorized decryption service. When Paillier encryption is employed, this approach supports operations such as adding encrypted values and multiplying an encrypted value by a publicly known scalar. However, the scope of encrypted computation must be carefully defined because practical tax calculations may involve income thresholds, comparisons, conditional deductions, and multiplication of private values. These operations are not all directly supported by Paillier's additive homomorphic property and may require alternative homomorphic encryption schemes or a hybrid processing approach. Consequently, the proposed system applies homomorphic encryption to selected tax-related computations rather than assuming that every tax calculation can be performed entirely on encrypted data.

System Architecture and Workflow:

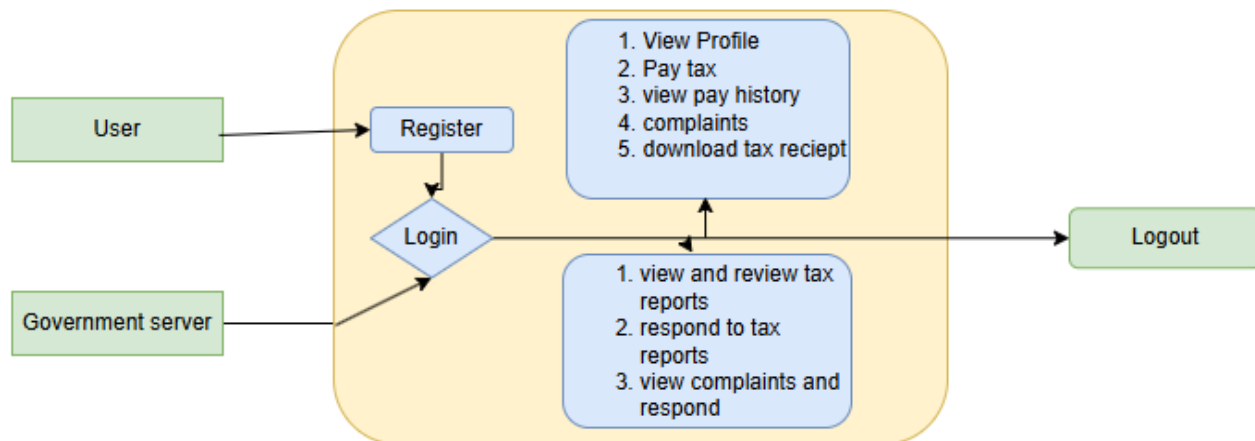


Figure: 1 Tax Management System Workflow Diagram

Figure 1 illustrates the proposed workflow of the privacy-preserving tax filing system. The process begins with taxpayer authentication through the web application, followed by the submission of tax-related information. The application validates the input data and converts it into an appropriate representation for cryptographic processing. Sensitive taxpayer attributes are then encrypted before being transferred to the homomorphic encryption (HE) computation service. The service performs only those operations supported by the selected encryption scheme and returns the computed results in encrypted form. These results are subsequently decrypted by an authorized key holder and presented to the appropriate user. The system also maintains filing status, payment references, and receipts for future access by taxpayers. Meanwhile, government officials can review reports and manage taxpayer complaints through a role-based administrative interface. The proposed prototype uses Python and Django for application development, with HTML, CSS, Bootstrap, and JavaScript supporting the user interface.

From an architectural perspective, the system separates the web interface, identity and access-control service, tax-rule engine, HE computation service, key-management component, payment and receipt subsystem, and government administration interface into distinct logical components. This separation is intended to improve maintainability, support controlled access to sensitive information, and establish clear security boundaries between application functions and cryptographic processing. However, the project documentation specifies that integration with external banking systems or payment gateways is outside the current scope. Therefore, payment-related functions in the prototype represent an application-level workflow rather than a verified external payment transaction. Actual payment processing would require the separate integration and testing of a suitable payment gateway.

Homomorphic Computation Model:

Let $Enc_{pk}(m)$ denote encryption of plaintext m under public key pk , and $Dec_{sk}(c)$ denote decryption under secret key sk . For an additive homomorphic scheme, an operation $\odot$ on ciphertexts satisfies:

$$Dec_{sk}(Enc_{pk}(m_1) \odot Enc_{pk}(m_2)) = m_1 + m_2$$

In the Paillier cryptosystem, multiplying two ciphertexts modulo n^2 produces an encryption of the sum of their corresponding plaintexts. Similarly, raising a ciphertext to a publicly known integer exponent k enables multiplication of the underlying plaintext by k . These properties make Paillier encryption suitable for privacy-preserving aggregation of financial amounts and the computation of linear expressions over encrypted data. However, correct implementation requires consistent plaintext encoding and careful management of numerical ranges to prevent overflow or modular wraparound (Paillier, 1999).

Nevertheless, tax computation involves more than simple addition or linear operations. Tax regulations may require selecting tax rates according to income brackets, applying deductions based on eligibility criteria, and evaluating conditional rules. Such operations involve comparisons, threshold checks, and conditional branching, which cannot be directly performed using Paillier's additive homomorphic properties alone. Supporting these requirements may necessitate a suitable homomorphic encryption

scheme capable of evaluating the required computations, secure multiparty computation protocols, or a hybrid architecture in which only explicitly authorized results are decrypted. Therefore, the encryption scheme, security parameters, numerical encoding method, rounding rules, and applicable tax-rule version must be clearly defined and validated before the correctness and reliability of the proposed tax computation process can be established.

Implementation Considerations:

The project documentation specifies Python 3.10 or later and the Django framework for server-side development, while HTML, CSS, Bootstrap, and JavaScript are used to build the user interface. The proposed prototype can be structured into modular components for user authentication, taxpayer profile management, tax filing, encrypted computation, payment-status tracking, receipt generation, filing history, complaint handling, and government administration. To improve flexibility and maintainability, the homomorphic encryption (HE) functionality should be encapsulated within a dedicated service interface. This modular approach would allow the underlying cryptographic library or encryption scheme to be upgraded or replaced without requiring extensive modifications to the overall web application.

The implementation should also incorporate essential security controls, including authenticated endpoints with role-based authorization, secure password hashing, Cross-Site Request Forgery (CSRF) protection, secure session management, rigorous input validation, and audit logging that excludes sensitive plaintext tax information. Additional safeguards should include encrypted backups, appropriate key-rotation procedures, and a documented key-recovery and disaster-recovery process. The private decryption key used by the HE component should be isolated from the application database and protected against access by unauthorized personnel, including ordinary system administrators. These measures represent recommended security requirements for a robust implementation; however, the available project documentation does not establish that they have been implemented or independently verified in the current prototype.

Results and Discussion:

The project presentation illustrates the intended functionality of the proposed tax filing system through various application screens, including the home page, services section, tax calculator, user registration and login pages, taxpayer dashboard and profile, tax payment interface, receipt generation, tax reports, issue reporting, government login, administrator dashboard, report review, and complaint management. These interfaces demonstrate the planned functional coverage of the prototype and its support for interactions between taxpayers and government officials. However, the available project materials do not provide the application source code, test cases, cryptographic parameter specifications, execution-time measurements, memory utilization data, throughput results, security audit findings, or comparative performance results against a conventional tax filing system.

Accordingly, the results are presented in terms of functional workflow coverage rather than as quantitative evidence of computational efficiency or verified security. A comprehensive evaluation would require measuring encryption and decryption latency, computation time for different operations and data sizes, memory consumption, storage overhead, and system performance under concurrent user loads. The correctness of encrypted computations should also be verified against a corresponding plaintext reference implementation. In addition, security testing should examine role-based access restrictions, malformed input handling, assumptions concerning private-key compromise, and the potential disclosure of sensitive taxpayer information through application logs, error messages, or other system outputs. These evaluations are necessary to establish the practical performance, computational correctness, and security properties of the proposed system before deployment in a real-world tax administration environment.

Limitations of the Work:

- The system does not define a formal threat model or clearly specify the roles of public-key holders, private-key holders, authorized decryption parties, and permitted outputs.
- The current implementation does not demonstrate complete encrypted evaluation of complex tax rules involving comparisons, tax-bracket selection, rounding, and conditional deductions.
- The prototype lacks quantitative performance benchmarks against conventional server-side tax computation using realistic synthetic taxpayer records.
- A formal cryptographic parameter review and independent security assessment have not been documented. Homomorphic encryption alone does not prevent metadata leakage, malicious inputs, or incorrect computation.
- The implementation does not establish whether advanced security measures, including multi-factor authentication, comprehensive audit trails, and robust backup and recovery mechanisms, are fully implemented and tested.
- External payment-gateway integration and comprehensive mobile accessibility have not been demonstrated in the current prototype.

Literature-Based Performances and Security Evaluation:

The project documentation does not provide empirical measurements of execution time, hardware configuration, benchmark datasets, system load-testing outcomes, or formal security-testing results. Consequently, the performance figures discussed in this section are derived from previously published research and serve only as background information for contextual comparison. They should not be interpreted as experimental findings obtained from the proposed Django-based prototype. Furthermore, performance results reported in different studies may not be directly comparable because they can vary significantly depending on the homomorphic encryption scheme, security parameters, hardware specifications, implementation techniques, and computational operations evaluated.

Reported Execution-Time Results:

A comparative study of partially, somewhat, and fully homomorphic encryption schemes published in 2025 reported Paillier and ElGamal processing times for batches containing 100, 1,000, and 10,000 messages. The reported measurements indicate that the computational cost of Paillier encryption increases considerably as the batch size grows within the evaluated implementation. These findings illustrate the potential computational overhead associated with processing encrypted data and highlight the importance of evaluating scalability when incorporating homomorphic encryption into privacy-sensitive applications. However, the reported timings reflect the specific experimental setup of the referenced study and do not directly represent tax calculation performance or the execution efficiency of the proposed Django-based tax filing system.

Scheme / batch	Encryption time (ms)	Decryption time (ms)	Ciphertext size (MB)	RAM (MB)
Paillier / 100	119	88	0.03	4.80
Paillier / 1,000	1,060	838	0.26	5.16
Paillier / 10,000	10,441	8,374	2.55	8.67
ElGamal / 100	57	11	0.03	4.10
ElGamal / 1,000	453	77	0.26	4.53
ElGamal / 10,000	4,565	680	2.56	8.13

Table 1: Comparative Performance Evaluation of Paillier and ElGamal Encryption Schemes Across Different Batch Sizes

A separate benchmark study on secure comparison protocols evaluated cryptographic operations under a 128-bit security setting using an Intel Core i5-9400F processor operating at 4.0 GHz. The study reported a Paillier encryption time of 22.98 ms, a decryption time of 7.756 ms, a key-generation time of 1,871 ms, and a ciphertext addition time of 0.021 ms. Although the study also reported lower execution times for its BFV-based configuration, these results must be interpreted in the context of the respective cryptographic schemes. BFV and Paillier differ in their mathematical foundations, parameter configurations, supported operations, and computational characteristics. Therefore, the reported measurements provide useful background on the performance of different homomorphic encryption approaches but should not be interpreted as a direct performance comparison or as evidence of the expected execution efficiency of the proposed tax filing system (A High Throughput BFV-Encryption-Based Secure Comparison Protocol, 2023).

Scheme (reported benchmark)	Key generation (ms)	Encryption (ms)	Decryption (ms)	Ciphertext addition (ms)
Paillier	1,871	22.98	7.756	0.021
BFV	0.360	0.459	0.055	0.002
Optimized BFV	0.198	0.251	0.060	0.002

Table 2: Comparative Cryptographic Operation Timings for Paillier, BFV, and Optimized BFV Encryption Schemes.

The Internet Society highlights that fully homomorphic encryption (FHE) can incur computational overhead several thousand times greater than equivalent plaintext processing, although the actual performance penalty varies considerably with the encryption scheme, computational task, and implementation. This estimate should be regarded as a general indication of the potential cost of homomorphic computation rather than a measured performance ratio for the proposed tax filing prototype. These considerations suggest that homomorphic encryption should be applied selectively to operations for which its capabilities are

appropriate, such as encrypted aggregation using an additive homomorphic scheme. A comprehensive performance evaluation of the specific tax-processing operations is necessary before drawing conclusions about computational efficiency, scalability, or application response time.

Interpretation for the Proposed Tax Filing System:

- The published benchmark results indicate that encryption and decryption may contribute substantially to overall processing latency as the number of taxpayer records increases. Therefore, the proposed system should evaluate these cryptographic operations separately from Django request processing, database interactions, and payment and receipt-management workflows.
- Although the cited comparison reports low computational cost for Paillier ciphertext addition, this does not necessarily imply that complete tax calculations can be performed efficiently. The overall processing time may also depend on input encryption, key management, data transfer, result decryption, and complex operations such as tax-bracket comparisons and conditional deductions.
- The project documentation describes several functional workflows, including tax filing, administrative report review, payment-status tracking, receipt downloads, and complaint management. However, the available benchmark studies do not evaluate these application-level functions and therefore cannot establish the end-to-end performance or transaction-processing capacity of the proposed tax filing system.
- A reliable performance evaluation should use synthetic taxpayer records and document the experimental environment, including CPU specifications, RAM, operating system, Python and Django versions, homomorphic encryption library, cryptographic key size or security level, batch size, and number of repeated trials. The evaluation should report mean execution times and standard deviations, compare encrypted processing with a conventional plaintext baseline, and verify that both approaches produce equivalent tax-rule outputs. It should also clearly identify which taxpayer attributes remain encrypted during computation and which outputs are revealed to authorized parties.

Literature-Informed Security Assessment:

Existing research and cryptographic standards indicate that homomorphic encryption (HE) can protect the confidentiality of encrypted inputs against a computation service that does not possess the corresponding decryption key, provided that the underlying scheme's security assumptions are satisfied and appropriate cryptographic parameters are selected. The Homomorphic Encryption Security Standard provides descriptions of supported encryption schemes and recommended security parameters to guide implementation decisions. Accordingly, a practical implementation should adopt parameters consistent with an applicable, up-to-date security standard and document the rationale for their selection rather than relying on arbitrary key sizes (Albrecht et al., 2018).

Nevertheless, homomorphic encryption alone does not guarantee that the server has executed the intended tax algorithm correctly, that the submitted taxpayer information is accurate, or that the resulting information is disclosed only to authorized parties. Verifying computational integrity and confirming that the intended function was executed are separate security requirements that may necessitate additional verification mechanisms. Furthermore, research has demonstrated that certain homomorphic encryption implementations may be susceptible to implementation-level side-channel attacks. Potential leakage through execution timing, memory-access patterns, error messages, and other observable behaviours must therefore be considered when evaluating the deployed system. These concerns represent risks identified through cryptographic research and are not confirmed vulnerabilities in the proposed prototype.

Security property / risk	What literature supports	Status for this project
Ciphertext confidentiality during supported computation	HE permits supported operations on ciphertexts without exposing plaintext to the evaluator, under scheme assumptions and secure parameters.	Design goal; no independent implementation audit supplied.
Parameter and scheme security	Security standards provide recommended parameter sets and explain scheme assumptions.	Must document selected scheme, library, parameters, and key lifecycle.
Correct execution / result integrity	HE alone does not guarantee that the evaluator ran the intended computation or that inputs are valid; verifiable computation is a separate research area.	Not demonstrated; needs correctness tests and possibly proofs or independent verification.

Side-channel and application leakage	Implementation-specific side channels and metadata leakage can undermine cryptographic confidentiality.	No penetration test or side-channel test report supplied.
Authentication and role authorization	HE does not replace account security, access control, secure sessions, or audit controls.	The slides describe user/admin login; control effectiveness has not been independently tested.

Table 1. Security Properties, Literature Support, and Implementation Status of the Proposed Homomorphic Encryption-Based Tax Filing System

Therefore, the proposed system should not be described as guaranteeing end-to-end security or completely preventing data breaches. Instead, it is more appropriate to state that the system architecture aims to minimize the exposure of sensitive plaintext information during selected computational operations through the application of homomorphic encryption. The actual security effectiveness of the prototype has not yet been independently established and requires further validation through formal threat modeling, detailed implementation review, and comprehensive security testing.

Recommended Evaluation Protocol for the Prototype:

- **Correctness Evaluation:** Validate the accuracy of homomorphic encryption (HE) computations by comparing decrypted results with those obtained from a trusted plaintext implementation for every supported tax operation. Testing should include boundary conditions such as zero values, maximum encoded values, rounding rules, and tax-bracket thresholds to identify potential computational discrepancies.
- **Performance Evaluation:** Assess key-generation time, encryption and decryption latency, homomorphic evaluation time, total application response time, throughput, peak memory consumption, and ciphertext size across multiple batch sizes. Conduct repeated experiments and report average results along with appropriate measures of variability to evaluate computational efficiency and scalability.
- **Security Assessment:** Examine key storage and access controls, session management, role-based authorization, input validation, application logs, backup files, and error-handling mechanisms. Independent penetration testing and an assessment of relevant side-channel risks should also be conducted to identify potential vulnerabilities in the implementation.
- **Reproducibility and Transparency:** Document the hardware and software configurations, homomorphic encryption library and version, selected encryption scheme and cryptographic parameters, synthetic dataset generation procedure, and testing scripts. Clearly distinguish the components and security properties that have been evaluated from those that remain untested to support reproducibility and enable independent verification of the results.

CONCLUSION:

This study proposes a privacy-preserving tax filing system that integrates taxpayer and government administration functionalities with homomorphic encryption (HE) for selected computational operations. Paillier additive homomorphic encryption provides a practical mechanism for performing encrypted summation and linear aggregation, whereas complex tax calculations involving comparisons, conditional deductions, and tax-bracket selection require additional cryptographic techniques or a carefully designed hybrid processing approach. The available project documentation outlines the intended system modules and interface functionalities but does not establish quantitative performance results or independently verified security guarantees. Further development should focus on defining a precise tax computation model, establishing secure key management and access-control mechanisms, and systematically evaluating computational correctness, performance overhead, and security against a clearly specified threat model. Such validation is essential before the proposed system can be considered suitable for deployment in a government tax administration environment.

REFERENCES:

- [1] Acar, A., Aksu, H., Uluagac, A. S., & Conti, M. (2018). A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys*, 51(4), Article 79. <https://doi.org/10.1145/3214303>.
- [2] Albrecht, M., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., Halevi, S., Hoffstein, J., Laine, K., Lauter, K., Micciancio, D., Moody, D., Morrison, T., Sahai, A., & Vaikuntanathan, V. (2018). Homomorphic encryption security standard. *HomomorphicEncryption.org*. <https://homomorphicencryption.org/standard/>
- [3] Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing* (pp. 169–178). <https://doi.org/10.1145/1536414.1536440>.

- [4] Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. In J. Stern (Ed.), *Advances in Cryptology—EUROCRYPT '99* (Lecture Notes in Computer Science, Vol. 1592, pp. 223–238). Springer. https://doi.org/10.1007/3-540-48910-X_16
- [5] A High Throughput BFV-Encryption-Based Secure Comparison Protocol. (2023). *Mathematics*, 11(5), 1227. <https://doi.org/10.3390/math11051227>.
- [6] A Comparative Study of Partially, Somewhat, and Fully Homomorphic Encryption in Modern Cryptographic Libraries. (2025). *Electronics*, 14(23), 4753. <https://www.mdpi.com/2079-9292/14/23/4753>.
- [7] Internet Society. (2023). Homomorphic encryption: What is it, and why does it matter? <https://www.internetsociety.org/resources/doc/2023/homomorphic-encryption/>