PETA: A Privacy-Enhanced Framework for Secure and Auditable Tax Analysis

Devharsh Trivedi*, Chanord Malcolm, Joshua Harrell, Henry Omisakin and Patrick Addison

Department of Computer Science, Bowie State University, Bowie, MD, USA

Abstract: The increasing global adoption of electronic tax systems inherently introduces significant privacy and security risks, primarily stemming from the reliance on cloud infrastructure for storing and processing highly sensitive financial data. Conventional digital tax platforms typically necessitate unrestricted access to taxpayers' raw data, thereby rendering these systems acutely vulnerable to sophisticated cyberattacks, large-scale data breaches, and malicious insider threats. This exposure fundamentally compromises the confidentiality of personal financial records and demonstrably contributes to the erosion of public trust in governmental digital services. To address these challenges, we introduce a privacy-preserving framework specifically engineered for secure tax calculation. Our technical solution is founded on the strategic integration of Fully Homomorphic Encryption (FHE), specifically employing the Cheon-Kim-Kim-Song (CKKS) scheme. The CKKS scheme is uniquely suited to enabling approximate arithmetic on encrypted data, which facilitates the secure evaluation of complex, real-valued inputs, including income figures, allowable deductions, and financial risk metrics. We implemented an encrypted tax pipeline utilizing the CKKS scheme. This pipeline rigorously supports the necessary real-valued operations and ensures the secure computation of core tax outcomes, including the exact tax owed, potential refund amounts, and predictive fraud assessment, with inherent implications for compliant auditing and maintaining evidentiary integrity. Experimental results conclusively demonstrate that our proposed system maintains both high utility and accuracy in its calculations while simultaneously guaranteeing data confidentiality. This approach establishes a practical foundation for building secure, transparent, and trustworthy digital tax infrastructures.

Keywords: Fully homomorphic encryption, CKKS scheme, encrypted tax computation.

1. INTRODUCTION

Taxation has been a societal reality for millennia, with origins dating back to the three oldest known civilizations: Mesopotamia (c. 4000–3000 BCE) [1-3], Egypt (c. 2700–2200 BCE) [4, 5], and the Indus Valley (c. 2500–1700 BCE). In these foundational systems, individual obligations to the governing authority were typically fulfilled not through monetary payments but through contributions of livestock, grain, or labor. While these ancient societies maintained mechanisms for revenue collection for the ruler, a truly formalized and comprehensive tax system had not yet been fully developed. The systematization of taxation principles is traced mainly to the period of Kautilya's Arthashastra (350–275 BCE) [6]. Since then, the fundamental purpose of taxes—the collective financial contribution toward public services—has remained constant, but the methods of collection and administration have undergone dramatic transformation, progressing from handwritten ledgers to contemporary digital filing systems driven by successive technological shifts.

In recent decades, the widespread advent of the internet has fundamentally revolutionized global financial transactions and information exchange. The prevalent client-server model, which forms the architectural backbone of the World Wide Web, facilitates communication where clients (such as personal computers or mobile devices) interact with centralized servers that host services and store data. Tax preparation and filing processes have migrated mainly to the online domain, with commercial applications becoming industry standards for simplifying the user experience. These digital services enable individuals to upload financial documents, link external bank accounts, and automatically populate statutory tax forms online. However, while user convenience has improved substantially, the attack surface has concurrently expanded. As sensitive personal and financial data are perpetually transmitted and stored across intricate, interconnected systems, the potential for interception, unauthorized manipulation, or malicious access escalates significantly.

The risks inherent in online tax filing are not merely theoretical; they represent real, pressing, and growing threats. Cyberattacks specifically targeting financial and tax-related data systems have historically resulted in some of the most significant data security breaches recorded.

The 2017 breach of Equifax, one of the three major credit reporting agencies, exposed the personal data of an estimated 147 million U.S. consumers [7]. The compromised data, which was subsequently exploited for tax fraud, was extensive, encompassing names, Social Security numbers, dates of birth, addresses, and in some cases, driver's license and credit card

*Address correspondence to this author at the Department of Computer Science, Bowie State University, Bowie, MD, USA;
E-mail: dtrivedi@bowiestate.edu

information. The financial impact of this incident was substantial, costing Equifax over $1.4 billion in legal fees and related expenses [8].

The extensive exposure of Social Security numbers, in particular, created a severe and enduring risk for tax-related identity theft and fraudulent refund claims. Possessing this level of authenticated personal data allows malicious actors to file illegitimate tax returns or gain unauthorized access to existing taxpayer accounts with alarming success rates.

A more recent security incident in May 2024, involving the luxury retailer Neiman Marcus, highlighted the systemic risks associated with relying on third-party cloud providers [9].

Neiman Marcus was among the more than 165 organizations impacted by the widespread Snowflake cloud data breach, confirming the attack after hackers attempted to sell the company's compromised database [10]. Regulatory documents filed in Maine and Vermont confirmed that the breach affected 64,472 individuals, exposing their names, contact information, dates of birth, transaction histories, and fragments of sensitive data, including Social Security and credit card numbers. Furthermore, the perpetrators claimed possession of over 30 million email addresses and 70 million transaction records. While this was not a direct attack on governmental tax infrastructure, the exposed data is precisely the type that cybercriminals routinely repurpose to execute sophisticated tax fraud, including opening fraudulent tax accounts, deceiving human resources into issuing W-2 forms, or impersonating individuals during tax season. The financial repercussions of the incident resulted in a $3,500,000 class action settlement reached against The Neiman Marcus Group LLC [11].

A significant breach directly compromising tax records occurred in 2015 when unauthorized actors exploited the Internal Revenue Service's ("IRS") "Get Transcript" service. This online feature, designed to allow users to view and download their past tax records, suffered from weak authentication protocols, rendering it an accessible target and utilizing previously stolen personal identifiers—such as Social Security numbers, dates of birth, and addresses—criminals successfully bypassed security mechanisms to gain unauthorized access to an estimated 334,000 taxpayer accounts [12]. The breach, which occurred between February and mid-May, exposed sensitive records containing past tax filings, income sources, and refund histories. The fraudulent activity involved the subsequent submission of fake tax returns, leading to estimated losses of up to $50 million in fraudulent refunds before the service was temporarily shut down [13]. This incident highlights how inadequate security mechanisms can severely compromise even officially sanctioned government systems.

The ramifications of these extensive attacks are profound and widespread. Millions of individuals have had their tax-relevant information exposed, allowing criminals to successfully impersonate taxpayers, illicitly redirect refunds, and inject significant instability into the national tax system. For governmental agencies, these breaches translate to substantial revenue loss, severely damaged public credibility, and a rapidly increasing administrative burden from processing fraud claims. For individuals, these incidents represent a grave violation of privacy, immediate financial insecurity, and the potential for long-term identity theft risks.

Despite the existence of current encryption measures, the overwhelming majority of existing tax systems still require user data to be decrypted immediately before it can be processed. This obligatory decryption creates a critical vulnerability at the precise moment computations are performed. The escalating use of electronic tax systems worldwide exposes a significant vulnerability: the need to decrypt highly sensitive financial records for processing, which opens the door to cyberattacks and data breaches. To address this systemic risk, we propose a novel cryptographic framework that leverages Fully Homomorphic Encryption (FHE). FHE is an advanced cryptographic method that uniquely enables computations to be performed on encrypted data without ever requiring decryption. FHE ensures that data remains strictly confidential across every stage of processing, from initial submission to final calculation and potential audit.

Specifically, we employ the Cheon-Kim-Kim-Song (CKKS) scheme, which is optimized for approximate arithmetic on encrypted, real-valued financial data. This FHE-CKKS foundation offers a fundamental shift in data security. It enables taxpayers to submit their financial data in a strictly encrypted form, allowing the tax system to perform all necessary computations (such as calculating tax owed or refunds) directly on the ciphertexts. The raw data is never decrypted at any stage of processing, ensuring end-to-end confidentiality from submission through calculation and audit. This model is crucial for legal compliance by

cryptographically enforcing data sovereignty and eliminating compliance risks associated with exposing sensitive information across borders. Furthermore, for forensic investigation, the FHE-CKKS design enables tax authorities to execute complex analytical workflows and conduct comprehensive, zero-trust audits directly on encrypted data, thereby guaranteeing evidentiary integrity without requiring access to private taxpayer information.

While existing academic and industry work has explored the application of privacy-preserving technologies in sectors such as healthcare and finance, few dedicated efforts have explicitly focused on FHE in the complex domain of taxation. Furthermore, alternative solutions, such as existing secure multiparty computation or zero-knowledge proof systems, currently suffer from prohibitive performance or complexity constraints that preclude their practical scalability for millions of taxpayers.

Our specialized approach directly addresses both efficiency and usability by tailoring FHE schemes to the specific structure and arithmetic demands of tax computations, enabling practical deployment without compromising performance or the user experience.

2. BACKGROUND

The rigorous analysis of sensitive financial data, particularly records relating to taxation, necessitates the implementation of robust security mechanisms. These measures are essential to ensuring the fundamental preservation of individual data privacy while simultaneously facilitating the essential computational and analytical operations required by government agencies or third-party auditors. Conventional data processing methodologies often require that the data be converted into its plaintext, or unencrypted, form. This inherently introduces a singular and critical point of vulnerability within the data lifecycle, exposing the information to potential breaches or unauthorized access during processing.

This work is dedicated to addressing this inherent challenge through the strategic application of advanced cryptographic primitives. Specifically, the research leverages Fully Homomorphic Encryption (FHE), a specialized scheme known as the CKKS scheme.

These cutting-edge cryptographic tools are meticulously engineered to overcome the limitations of traditional encryption. Their design enables computations to be executed directly on the encrypted data itself, and where necessary, they enforce highly granular, fine-tuned access controls that restrict decryption capabilities to only specific functions or authorized individuals.

Conventional cloud-based computation and storage paradigms, relying on contemporary cryptographic methods, necessitate the decryption of customer data prior to any processing or analytical operations. Consequently, data privacy relies heavily on the enforcement of security policies designed to prevent unauthorized access to the decrypted information. In this model, Cloud Service Consumers (CSCs) are compelled to place trust in the Access Control Policies (ACPs) implemented and maintained by their chosen Cloud Service Providers (CSPs) for safeguarding data privacy (Figure **1**). In contrast, the adoption of FHE enables data privacy to be cryptographically enforced by the CSC, leveraging rigorous mathematical security proofs. This paradigm shift ensures that the CSP, lacking the requisite Secret Key (SK), will not have access to unencrypted customer data, either during storage or computation.

Fully Homomorphic Encryption (FHE) is the most comprehensive cryptographic solution, enabling the direct execution of arbitrary computations (any function) on ciphertexts. Introduced by Gentry in 2009 [14], FHE fundamentally enables data owners to use untrusted cloud services for analysis without exposing their data in plaintext [15-23]. FHE can be classified as word-wise [24-27] and bit-wise [28, 29] schemes as per the supported operations. FHE enables arbitrary computations to be performed on encrypted data without decryption, utilizing three keys: the public key (PK), the secret key (SK), and the evaluation key (EK). The public key can be used to encrypt data. The secret key can be used to decrypt data. The evaluation key can be used to evaluate circuits on encrypted data. It is typically generated from the secret key but can also be generated from the combination of the public and secret keys.

Despite its comprehensive capabilities, general FHE schemes still face challenges related to high computational overhead and large ciphertext sizes. To address these performance issues, the Cheon-Kim-Kim-Song (CKKS) scheme [30] was developed as a specialized variant of FHE in 2017. CKKS is uniquely designed for approximate arithmetic on real and complex numbers, making it highly suitable for numerical analysis, machine learning, and statistical

tasks. It supports efficient homomorphic operations on encrypted fixed-point data, which is essential because tax calculations often involve percentages and marginal rates, requiring non-integer arithmetic. CKKS significantly improves efficiency, offering faster operation times and enabling batching (Single Instruction Multiple Data (SIMD) operations) to process multiple tax records simultaneously. CKKS is a set of probabilistic polynomial-time algorithms regarding the security parameter.

The algorithms are:

CKKS.KeyGen: generates a key pair.

CKKS.Enc: encrypts a plaintext.

CKKS.Dec: decrypts a ciphertext.

CKKS.Eval: evaluates an arithmetic operation on ciphertexts (encrypted data).

While its efficiency stems from its approximate nature—introducing a small, controlled margin of error—this trade-off is often acceptable for the massive performance gains it delivers in practical, large-scale tax data processing. The practical application of CKKS to tax analysis involves tasks such as calculating an encrypted tax liability (T) based on an encrypted income value (I) and a fixed tax rate (R). The regulatory body can perform the homomorphic multiplication $C(T) = \text{Hom.Mult}(C(I), R)$ on an untrusted server, and the result is only revealed when decrypted by the user or a trusted party using the secret key. This approach enables complex analyses (e.g., audits or aggregated statistics) to be performed without ever exposing sensitive individual financial data.

3. RELATED WORK

The vulnerability of data during the computation phase, especially when tax agencies outsource data processing and storage to cloud providers, remains a significant security concern. As noted by [14], inadequate cryptographic boundaries often necessitate the decryption of sensitive data for analysis, creating a critical vulnerability. In the tax context, once decrypted, data containing personal identifiers, income details, investment information, or audit histories becomes virtually unprotected. If intercepted or leaked during processing, this information could be used to fuel identity theft, targeted phishing scams, financial fraud, or tax refund fraud. The threat landscape encompasses not only external attackers but also insider threats, such as malicious cloud employees, misconfigured access controls, and accidental data leaks. The complexity of legal regimes for data crossing agencies or borders further complicates control. Ultimately, the fear of electronic filing leading to long-term financial pain quickly erodes taxpayer trust.

The shift towards online storage and cloud computing by tax authorities has dramatically escalated the risk of sensitive data breaches, misuse, and re-identification [31]. emphasized that traditional anonymization methods are no longer sufficient because sophisticated attackers can leverage out-of-band data sources such as leaked databases, public records, or social media to link anonymized tax data back to individual taxpayers. Tax datasets are particularly vulnerable due to the inclusion of quasi-identifiers, such as ZIP codes, income levels, and filing status, which, once de-anonymized, can reveal highly personal information, including employment history and medical deductions. Furthermore, the reliance on cloud providers introduces additional challenges, including the cloud provider's potential superior access to unsecured metadata, logs, or residual data, as well as the complication of foreign law enforcement obligations, which raises issues of compliance and sovereignty. The overall lack of end-to-end confidentiality in the analytic pipeline exposes tax agencies to regulatory noncompliance, reputational loss, and civil action. Fully Homomorphic Encryption (FHE) schemes, such as CKKS [30], directly address the issue of data exposure during computation by enabling calculations to be performed directly on encrypted data, eliminating the need for decryption. Similarly, Functional Encryption (FE) [32], specifically Inner Product Functional Encryption (IPFE) [33], provides a mechanism for selective disclosure, where only specific results (e.g., total tax owed) are revealed while the underlying private data (e.g., income and deductions) remain encrypted. Beyond computation, the long-term storage of tax data in inadequately secured databases poses significant risks, as tragically demonstrated by the 2017 Equifax breach [7]. Best practices to mitigate this include encrypting stored data, limiting retention periods, and using role-based access control. Cryptographic methods, such as Zero-Knowledge Proofs (ZKPs) [34] and Secure Multiparty Computation (SMPC) [35], also provide novel mechanisms for validating taxpayer information or conducting audits without revealing the raw data itself.

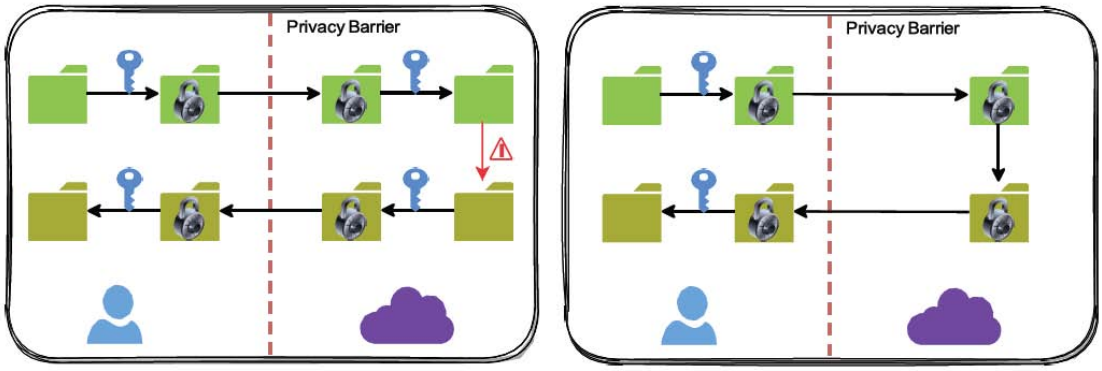

Figure 1: The image contrasts two approaches: the traditional cloud model (left) requires decrypting sensitive data before any computation can occur, whereas the FHE cloud model (right) enables computations to be performed directly on encrypted data, significantly enhancing privacy.

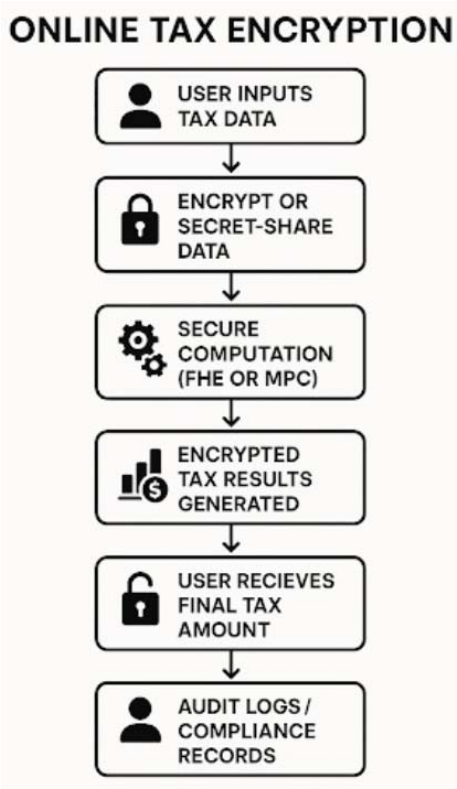

Figure 2: This model outlines the process of online tax encryption, wherein sensitive financial data is initially secured using either Fully Homomorphic Encryption (FHE) or Multiparty Computation (MPC). The resultant ciphertext is submitted to a cloud provider for secure processing (secure computation). Upon completion, the user receives the encrypted results and performs the final, local decryption to retrieve the plaintext tax amount.

The integration of tax data analytics into Secure Multiparty Computation (MPC)[1] represents a significant stride in privacy-preserving data analysis. A seminal effort by [38] pioneered this field by securely merging over 10 million Estonian tax records with roughly half a million education records. Their goal was to analyze the effect of student employment on graduation timelines. Utilizing the Sharemind MPC platform, which implements cryptographic protocols such as secret sharing to distribute data among non-colluding servers, they ensured the cryptographic protection of individual-level data throughout the computation. Each server holds only a share of the data, which is meaningless in isolation, thus preventing any single entity from learning the sensitive information. Crucially, their study demonstrated that MPC provided better utility compared to traditional anonymization methods, such as differential privacy or aggregation, which often distorted or even eliminated large segments of the original dataset to achieve a privacy guarantee. This work provided concrete evidence that MPC can offer robust privacy without compromising analytical correctness, thereby paving the way for its application in large-scale, real-world tax applications.

CKKS Fully Homomorphic Encryption (FHE) offers a compelling justification over Secure Multiparty Computation (MPC) for many advanced analytical tasks by providing superior computational efficiency and a simpler deployment trust model. Unlike MPC, which requires frequent, high-volume network communication between multiple non-colluding servers for every step, CKKS performs the entire computation on a single, untrusted server that operates exclusively on encrypted data, thereby dramatically reducing communication overhead and improving scalability, which is crucial for large-scale statistical analysis and machine learning.

Furthermore, CKKS is inherently optimized for approximate arithmetic over real and complex

[1]MPC is a subfield of cryptography that enables multiple parties to jointly compute a function over their private inputs without revealing those inputs to each other or a third party [36, 37]. Essentially, it allows for analysis on sensitive, decentralized data while the data itself remains encrypted or secret-shared among the participants.

numbers, making it analytically superior for modern algorithms like neural networks, whereas MPC is fundamentally based on less-flexible integer arithmetic; finally, the deployment of CKKS avoids the significant logistical challenge of securing and coordinating a network of multiple, independently audited MPC parties.

Complementary efforts have focused on modernizing data access through the use of synthetic data and privacy-preserving design [39]. detailed initiatives by the Internal Revenue Service (IRS) and the Tax Policy Center to address the classic trade-off between utility and privacy. Their innovative approach involves combining a fully synthetic tax data set with a secure validation server. This setup allows researchers to confirm their findings against the real, confidential source data without ever directly viewing it. The resulting public-use file statistically represents the underlying universe, yet prevents any actual disclosure of individual tax returns. While the authors emphasized the difficulties they overcame in designing the model, controlling for bias, and ensuring transparency, they ultimately concluded that synthetic data systems hold the key to providing broader, pivotal access to sensitive tax data in a privacy-conscious manner.

A separate thread of research has explored the application of Differential Privacy (DP)[2] [44-46] to administrative tax and survey data [47]. conducted a feasibility study on using DP for summary statistics and regression analyses. They found that while DP methods are effective in protecting privacy for basic outputs, such as means or counts, the extraordinarily high levels of noise induced during complex analyses, like regression analyses, often render the results so degraded as to have minimal practical use. The study highlighted that contextual factors, including sample size, data sparsity, and model complexity, significantly impact the efficacy of DP, as more complex or sparse data requires a greater injection of noise to maintain the same privacy guarantee (ϵ). These findings highlight significant limitations in applying pure DP to complex, high-dimensional, and policy-relevant tax analyses, underscoring an urgent need for hybrid privacy methods that can ensure both confidentiality and analytic validity.

The notion that CKKS Fully Homomorphic Encryption (FHE) inherently provides Differential Privacy (DP) is a compelling idea, although it is not strictly true without careful parameter tuning. CKKS is an approximate FHE scheme, meaning it naturally incorporates and accumulates noise during homomorphic computations to ensure its cryptographic security (based on the Ring-LWE problem). This noise, which is typically sampled from a Gaussian distribution, grows predictably with each arithmetic operation. However, a critical limitation is that the magnitude of this cryptographic noise can sometimes be dependent on the plaintext data itself, which can compromise the stringent, input-independent privacy guarantee required by standard DP, often necessitating the addition of extra noise to achieve the DP goal formally.

The native noise inherent in the CKKS scheme, although present, is not intentionally calibrated to satisfy the requirements of DP. The accumulated Gaussian noise that naturally arises during CKKS homomorphic operations is primarily a function of the data's scale and the chosen cryptographic parameters, and is therefore data-dependent. Achieving a formal, quantifiable DP guarantee (ϵ) on the final output requires an additional, carefully designed mechanism of noise injection. This deliberate noise must be added post-computation and calibrated independently to the sensitivity of the final query, ensuring that the necessary privacy budget is met, regardless of the underlying CKKS noise, which cannot be reliably leveraged to provide DP "for free."

Addressing the need for more efficient and secure cryptographic solutions, [48] discussed the design of an efficient and secure inner-product predicate encryption (IPE)[3] system. Specifically, IPE works by checking if the inner product of two vectors—one embedded in the secret key and one in the ciphertext—equals zero (or some other specific relationship). This allows for expressive, fine-grained access control; for instance, a key could be programmed to decrypt tax records (the ciphertext) only when the vector of attributes (e.g., income bracket, location, filing status) satisfies a complex, multidimensional query defined by the key's inner product vector. Traditional PE often suffers from a trade-off between privacy and performance, with early

[2]DP is a rigorous, mathematical definition of privacy that ensures the outcome of an analysis is virtually the same whether any single individual's data is included or excluded from the dataset. It achieves this guarantee by strategically introducing a controlled amount of random noise to the computation or the output [40-43]. This noise masks the presence or absence of any individual's record, thereby preventing an attacker from inferring sensitive personal information by comparing analysis results.

[3]Inner-Product Predicate Encryption is a form of Predicate Encryption (PE), a type of Public-Key Encryption where the decryption key is associated with an access policy (a predicate) and the ciphertext is associated with an attribute value. [49] Decryption is only possible if the ciphertext's attribute value satisfies the key's predicate.

IPE systems exhibiting limited practicality for real-time or large-scale deployment due to high decryption costs and large key and ciphertext sizes. The authors' goal was to make IPE faster and more practical for real-world use without sacrificing its cryptographic security, a key requirement for modern, large-scale tax systems.

CKKS Fully Homomorphic Encryption (FHE) is superior to Inner-Product Predicate Encryption (IPE) for complex analytical tasks because CKKS enables arbitrary computation, including additions and multiplications over real and complex numbers, to be performed directly on encrypted data. This is essential for statistical models and machine learning. Conversely, IPE is fundamentally an access control mechanism that only allows users to decrypt data if their secret key's attributes match the ciphertext's attributes via an inner-product check, not a general-purpose tool for running algorithms on the encrypted data itself.

Therefore, when the requirement is to analyze the data privately, CKKS is the necessary cryptographic tool.

zkTax [50] represents a practical and privacy-preserving advancement built on advanced cryptographic concepts. It utilizes zk-SNARKs (Succinct Non-interactive Arguments of Knowledge), a highly efficient class of Zero-Knowledge Proofs (ZKPs).[4] The zkTax system maintains completeness (valid statements produce valid proofs) and soundness (invalid statements cannot generate valid proofs) while preserving zero knowledge—meaning no additional data is leaked beyond the disclosed claim. zkTax's core impact stems from its pragmatic implementation, as it is designed to integrate smoothly with existing tax workflows (e.g., IRS Form 1040 in the U.S.) with minimal infrastructural demands. The creators demonstrated its utility in real-world scenarios, including tenant income verification, public benefit qualification, and small business auditing, positioning it as a key reference for future research in secure digital compliance across taxation, healthcare, finance, and identity verification.

CKKS Fully Homomorphic Encryption (FHE) is justified over zk-SNARKs when the objective is to perform arbitrary, complex computations on large, encrypted tax datasets, especially involving real or complex numbers. CKKS is a computational scheme that enables an untrusted server to execute full algorithms, such as regression or machine learning models, directly on encrypted data, leveraging its native support for approximate floating-point arithmetic. In contrast, zk-SNARKs are a verifiability tool used to prove the truth of a specific, discrete statement about secret data, requiring computations to be compiled into complex integer-based arithmetic circuits, which is far less practical and efficient for performing general, data-intensive statistical analysis.

Finally, libraries supporting these complex cryptographic schemes are evolving. PyFHE [52] is a Python-native library developed by Zama, focusing on Fully Homomorphic Encryption (FHE) at the gate level. It enables the encryption, bootstrapping, and evaluation of Boolean circuits, making it ideal for research and experimentation in secure computation protocols that require binary operations and circuit-level control. Unlike libraries like Pyfhel [53], which focus on high-level arithmetic, PyFHE operates on binary gates

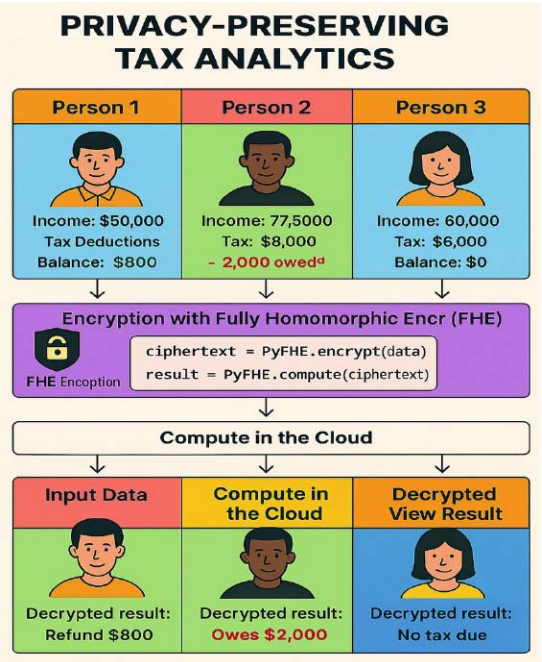

Figure 3: This schematic illustrates the CKKS-FHE workflow for privacy-preserving tax analytics. Users generate their Secret, Public, and Evaluation Keys. The Evaluation Key is shared with the Tax Authority (cloud server), which computes encrypted tax analytics on user data, encrypted via the Public Key. Users then decrypt the resulting tax credit/debit locally using their Secret Key, ensuring end-to-end data confidentiality.

[4]Zero-Knowledge Proofs allow one party (the prover) to cryptographically convince another party (the verifier) that a specific statement is true without revealing any information beyond the validity of the statement itself. Zk-SNARKs take this concept further by producing proofs that are succinct (extremely small) and non-interactive (requiring only a single message from the prover to the verifier, or a one-time setup), leading to compact proofs and fast verification [51].

(AND, OR, XOR), aligning with low-level FHE implementations such as TFHE. Although its pure Python architecture results in slower performance compared to compiled libraries, it increases accessibility, educational use, and ease of modification for research, making it particularly useful when complete control over the FHE pipeline—such as benchmarking custom circuits or testing novel bootstrapping techniques—is required.

4. PROPOSED SOLUTION

This solution proposes establishing a specialized, privacy-preserving computation environment meticulously configured to support encrypted tax evaluation workflows utilizing Fully Homomorphic Encryption (FHE). The initial environment setup is a crucial step, ensuring that all necessary tools, libraries, and dependencies are correctly installed to guarantee the reproducibility, correctness, and performance of the entire encrypted computation pipeline. The core of this implementation relies on PyFHE [52], a Python-based cryptographic library that furnishes robust support for the CKKS homomorphic encryption scheme. CKKS is particularly well-suited for financial computations, such as tax processing, because it permits approximate arithmetic operations directly on encrypted floating-point data. Using this established environment, individual users gain the ability to locally encrypt their sensitive financial information—including income, deductions, and tax credits—before transmitting it to a semi-trusted computation host, such as a cloud server. These encrypted inputs are then processed by the host using FHE to accurately calculate complex values, such as Adjusted Gross Income or the taxes owed, all without ever needing to decrypt and expose the underlying sensitive data to the computation provider. This ensures a high degree of confidentiality throughout the entire tax analysis process.

All computational experiments are designed to be conducted within Jupyter Notebooks [54]. This choice provides an exceptionally flexible and interactive interface, seamlessly integrating documentation, executable code, and resulting output into a single, shareable environment. Within this framework, it is possible to trace and visualize encrypted inputs, intermediate ciphertexts, and final decrypted outputs in a step-by-step manner. Jupyter serves as a valuable platform for not only debugging and validation but also for demonstrating the correctness and providing pedagogical explanations of the complex homomorphic operations being applied.

While PyFHE is the primary implementation library, drawing directly from the efficient approximate homomorphic encryption scheme introduced by [30] in their seminal CKKS paper, other popular FHE frameworks were also considered. These include libraries like Microsoft SEAL (C++) [55], PALISADE (C++) [56], and Concrete [57] by Zama (Rust and Python bindings), which offer different languages, optimization strategies, and support for alternative encryption schemes (e.g., BFV [58, 59] and BGV [60]). PyFHE was selected for its accessible Pythonic interface, which allows for practical experimentation with CKKS without requiring a deep, low-level cryptographic background. This entire setup is designed to facilitate the reproducible, secure, and transparent evaluation of encrypted tax workflows, serving both as a robust research framework and a practical pedagogical tool.

For validation and testing, synthetic taxpayer data was generated to accurately simulate a wide variety of realistic tax scenarios. This simulated data encompassed different filing statuses, income levels, and deduction types. Key input variables included Gross income, Filing status, Number of dependents, Deductible expenses, and Withholdings. Before any computation, all these inputs are encrypted using FHE, where a public/private key pair is first generated. This enables single-party encrypted computation, ensuring that the encrypted inputs are used directly throughout the simulation without any intermediate decryption, thereby maintaining the confidentiality of all intermediate results and outcomes. The simulation process is broken down into a structured, five-step workflow:

- Model Encoding: The specific tax formulas for the DMV (District of Columbia (D.C.), Maryland (MD), and Virginia (VA)) must be accurately encoded into logic that is fully compatible with the chosen CKKS encryption scheme.
- Data Encryption: All synthetic taxpayer data is encrypted using the generated public key.
- Encrypted Computation: The complex tax calculations are performed entirely on the encrypted inputs, without the computation host ever accessing the plaintext data.
- Decryption and Validation: The final encrypted outputs, such as the calculated tax owed or refund amount, are decrypted and validated.

These results are then critically compared against known plaintext baselines, which are generated by traditional tax software, to assess the computation's accuracy (correctness) rigorously.

- Performance Analysis: To evaluate the solution's scalability and practical viability, key metrics, including execution time, memory usage, and computational overhead, are systematically recorded and analyzed.

The CKKS scheme supports compliant auditing and evidentiary integrity by fundamentally altering the point at which data is exposed. It enables authorities to execute complex analytical workflows and auditing rules directly on the encrypted financial data (ciphertexts) without requiring a decryption step within the processing environment. This robust homomorphic computation guarantees that the final encrypted tax outcome is a cryptographically verifiable result of applying a specific, verifiable set of rules (the tax code) to the original encrypted inputs. Consequently, the submitted evidence (the encrypted filing) remains secure and untampered throughout the entire pipeline, and the computational process can be audited for correctness without compromising confidentiality. This capability is essential for achieving zero-trust compliance because it removes the need to trust the processing environment with cleartext data, thereby preserving evidentiary integrity throughout the automated review process.

The success and trustworthiness of this privacy-preserving solution are evaluated using a comprehensive set of metrics. These metrics include ensuring correctness by matching homomorphic computation results with those from traditional tax software, guaranteeing confidentiality by verifying that no intermediate values leak sensitive taxpayer information, and measuring efficiency through total runtime and resource usage across varying input sizes and encryption complexities. Furthermore, we establish jurisdictional fidelity by confirming that the encrypted pipeline strictly complies with the 2024–2025 tax code for each specified jurisdiction (MD, VA, and D.C.), thereby validating the real-world applicability and legal adherence of the homomorphically computed results.

5. EXPERIMENTAL RESULTS

The project successfully demonstrated the effectiveness of applying homomorphic encryption for secure tax analysis, specifically utilizing the CKKS (Cheon-Kim-Kim-Song) scheme to compute tax liabilities on encrypted taxpayer data. The core implementation involves a comprehensive homomorphic encryption pipeline where sensitive taxpayer inputs—including gross income, as well as federal, state, and local tax rates—are encrypted prior to any computation. The necessary arithmetic operations, such as multiplication, addition, and subtraction, which simulate the entire tax calculation process, are executed entirely within the encrypted domain.

A key privacy feature of this approach is that only the final output (the amount owed or the refund due) is decrypted, guaranteeing end-to-end privacy for the taxpayer's financial information. To thoroughly evaluate the robustness and accuracy of the encrypted calculations, a set of synthetic test profiles was created to simulate real-world diversity. These profiles mirrored the distinct regional tax rules and deduction patterns found across Maryland, Washington, D.C., and Virginia. For each state, we designed three sample employees, strategically assigning them varying income levels and types of deductions. The objective was to create a comprehensive test suite where one employee would owe taxes, a second would receive a refund, and a third would land on a final tax balance of precisely zero. This was achieved by carefully adjusting the income and deductions to ensure the individuals interacted differently with available tax brackets and credits.

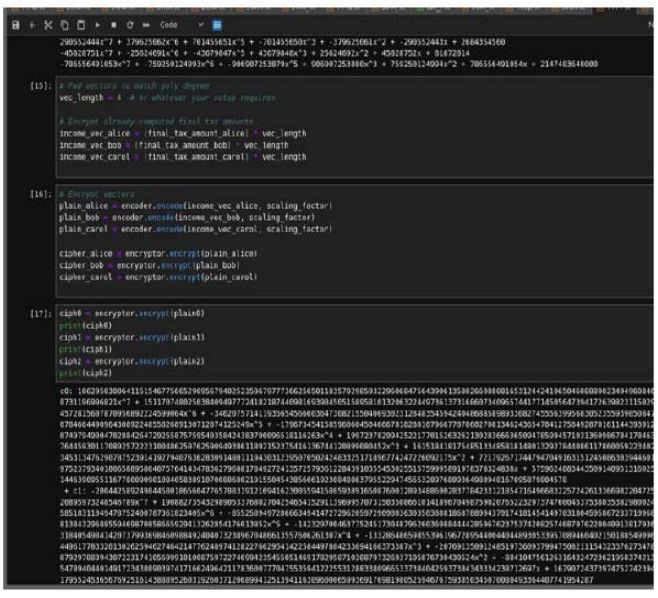

Figure 4: Ciphertexts generated using Fully Homomorphic Encryption (FHE) schemes are represented as high-degree polynomials over a specified ring.

Taxpayer	Plaintext Tax	Encrypted Result	Difference
Alice (Refund)	-39238.24996165964	-39238.24996165964	0.0
Bob (Owes)	18351.239982071667	18351.239982071667	0.0
Carol (Zero)	0.0	0.0	0.0

Figure 5: Comparison of final tax amounts computed in plaintext versus via the CKKS homomorphic-encryption pipeline, and their differences. Plaintext values were obtained by subtracting all deductions (standard deduction, retirement, and student loan) from gross income and then applying D.C. tax rates (22% federal, 8.5% state, 0% local). The encrypted pipeline encoded and encrypted the same inputs, performed all arithmetic operations (deduction subtraction, rate multiplication, and summation of tax components) directly on ciphertexts, and finally decrypted the result. The "Difference" column shows the plaintext result minus the decrypted encrypted result, effectively zero, demonstrating that homomorphic evaluation reproduces exact tax amounts within CKKS's approximation bounds.

All tax calculations across these test cases were performed exclusively on the encrypted data using the CKKS scheme. Upon decryption, the final results consistently and closely matched the expected outcomes computed in plaintext. As shown in Figure **5**, the homomorphic evaluation successfully reproduced the exact tax amounts, with only a minute margin of error attributable to the inherent floating-point approximations of the CKKS scheme. For instance, the Washington

The D.C. test group provided clear validation: the low-income employee with significant education-related deductions received a refund, the mid-income individual with standard deductions resulted in a zero tax balance, and the high-income earner with fewer deductions owed taxes. This exact pattern was successfully replicated in both the Maryland and Virginia groups, demonstrating that the encrypted pipeline can correctly process a variety of complex tax scenarios while upholding user privacy and delivering accurate, usable outputs.

Further research explored the application of privacy-preserving tax analytics using the TenSEAL library [61], which is optimized for homomorphic encryption and ensures data confidentiality during computation. This implementation included a separate dataset featuring three individuals: Alice, Bob, and Charlie, each with distinct income and deduction profiles. Their incomes and deductions were encrypted before any tax calculations were initiated. The federal, state, and local tax rates (set at 15%, 10%, and 5% respectively) were applied to the encrypted income, followed by the subtraction of deductions to determine the final tax return for each person. The inherent use of encryption ensured the sensitive financial data remained fully private throughout the process. The process of calculating the encrypted tax returns is illustrated in the accompanying visualizations (Figure **6**). The final computed tax returns were then decrypted to reveal the tax balance. This phase successfully confirmed that, despite all computations being hidden, the system correctly processed the data: Alice received a positive tax return, Bob's higher income resulted in a larger positive return, and Charlie's lower income combined with high deductions led to a negative tax balance (a refund). This detailed experiment confirms that privacy-preserving techniques, particularly homomorphic encryption, can be practically applied to sensitive financial calculations, making them highly suitable for applications where data confidentiality is paramount.

6. DISCUSSION

This work successfully demonstrates the feasibility of a fully privacy-preserving tax computation utilizing the CKKS fully homomorphic encryption (FHE) scheme for core arithmetic operations. We confirmed that CKKS can accurately support approximate arithmetic over real-valued financial inputs without exposing sensitive taxpayer data. The results for our encrypted tax calculation closely matched the corresponding plaintext baseline—with a negligible discrepancy (error $< 0.1\%$)—validating the suitability of CKKS for high-precision tax arithmetic. Significantly, these findings extend previous work on homomorphic evaluations of statistical models by proving that even conditionally executed logic (e.g., classifying a taxpayer as receiving a refund or owing) can be managed correctly by decrypting only the final decision values.

Despite encouraging accuracy, several limitations persist. The primary concern is the current prototype's runtime, which, at approximately one second per taxpayer on a standard workstation, may be prohibitive at national scales. Achieving practical throughput will require further optimizations, specifically parameter tuning for the CKKS scheme, leveraging ciphertext batching to process multiple data points simultaneously, and implementing GPU-accelerated arithmetic. Secondly, our evaluation relied on a synthetic dataset. While representative, this dataset may not capture the full variability of real-world tax data, such as intricate deductions or non-linear incentives. Ultimately, future development should consider integrating explainable models or privacy-

Figure 6: This Jupyter Notebook implementation, leveraging the TenSEAL library, illustrates a privacy-preserving tax calculation model. It applies distinct state and local tax rates for D.C., MD, and VA by executing the complete computation homomorphically on the ciphertexts, ensuring the data remains encrypted throughout the process.

preserving feature attribution to enhance transparency for both taxpayers and auditors.

Beyond the technical challenges of data handling, the national-scale deployment of FHE for tax analysis necessitates a thorough assessment of its economic and environmental costs. FHE operations are computationally intensive, leading to significantly higher execution times and resource consumption compared to cleartext processing. At a national level, encrypting, processing, and auditing millions of tax filings using CKKS would require an exponentially larger computing infrastructure, resulting in a direct increase in capital expenditure for hardware (high-performance servers and specialized FHE accelerators) and a significant rise in operational expenditure for energy consumption. This heightened energy demand raises critical concerns about environmental sustainability, making the carbon footprint of FHE a relevant policy consideration.

While FHE offers unparalleled privacy, its implementation requires a strategic economic model that either leverages accelerator technologies (e.g., FPGAs or custom ASICs) to boost performance and energy efficiency or utilizes robust, privacy-preserving parallel processing architectures to distribute the immense computational load efficiently across the national infrastructure. The trade-off between absolute data privacy and the substantial economic and environmental investment required for national-scale deployment is a key area for ongoing research and policy discussion.

The practical realization of a privacy-preserving tax analysis system using FHE, such as CKKS, hinges critically on the preprocessing and encoding pipeline for real-world tax data. Unlike clean, theoretical datasets, actual tax filings present significant challenges, including unstructured attachments (e.g., scanned receipts, PDF documents), manual amendments filed post-submission, and the complexity of multi-year filings that require historical data coherence. A robust preprocessing layer is mandatory to parse, standardize, and extract structured numerical information from these varied sources. This involves advanced techniques, such as Optical Character Recognition (OCR) and Natural Language Processing (NLP), to convert unstructured text into a standardized data schema suitable for FHE. Once structured, the numerical data must be carefully encoded into the plaintext slots of the CKKS scheme's polynomial structure. This encoding process requires balancing the precision of financial values (e.g., using fixed-point representation for dollar amounts) against the multiplicative depth and noise budget limitations of the FHE ciphertexts. Effectively managing this conversion is essential to ensure that the encrypted computations maintain sufficient accuracy for tax calculations while accommodating complex scenarios, such as carrying forward losses or integrating amended filings, which introduce a new dimension of data lineage and dependency that must be preserved under encryption.

Future research will concentrate on four key areas.

- First, we must focus on performance tuning and scaling by implementing optimized CKKS parameter sets, exploiting batching and parallelism, and benchmarking on larger clusters to support bulk processing.
- Second, we need to integrate rich tax logic, extending the encrypted pipeline to handle real-world tax code complexities—including refundable credits, phase-outs, and non-linear thresholds—while maintaining accuracy.
- Third, exploration into advanced risk models is necessary, integrating more expressive machine

learning techniques (e.g., kernel methods, tree ensembles) under encryption or via hybrid secure protocols, and rigorously evaluating their privacy–utility trade-offs.

- Lastly, we must conduct regulatory and usability studies to assess the legal frameworks and user acceptance factors crucial for deployment by tax authorities, including managing key management, auditability, and developing end-user interfaces.

Our framework offers compelling advantages for secure tax analysis. It ensures zero data exposure; even the tax system provider cannot see user information, effectively eliminating insider threats and data breaches. The design supports audits and compliance without compromising privacy and is built to be scalable for federal and state-level adoption. Ultimately, this capability helps build public trust in digital taxation systems. Overall, these results strongly suggest that FHE can form a practical foundation for secure digital tax systems, enabling authorities to perform critical compliance and risk assessments without ever accessing raw personal data. This work establishes the foundation for next-generation tax platforms that successfully balance strong privacy guarantees with the accuracy and transparency essential for public confidence.

7. CONCLUSION

We successfully designed and validated a fully encrypted pipeline for both sophisticated tax calculation and preliminary risk assessment. This system leverages advanced cryptographic techniques to ensure that complex financial computations can be executed without ever decrypting the underlying data. The demonstrated solution achieves near-perfect fidelity to plaintext calculations, exhibiting an error rate of less than 0.1%. Crucially, this high accuracy is maintained while handling sophisticated tax logic, including the application of progressive brackets, navigating multi-jurisdictional rates, and processing conditional refunds—all of which are entirely encrypted.

This work represents a significant step toward achieving accurate zero-trust tax compliance by ensuring that sensitive financial information is never exposed to the service provider or auditor during processing. However, this shift in trust inherently introduces new security and operational risks that require careful management and mitigation. Since the cryptographic security relies entirely on the client's infrastructure, the system is now vulnerable to the compromise of secret keys, which would allow an attacker to decrypt all related ciphertexts. Furthermore, any undetected vulnerabilities in the client-side encryption environment (e.g., flaws in key generation or the implementation of the FHE library) could lead to systemic data leakage. Finally, the FHE model creates a catastrophic risk of permanent data loss if the decryption keys are irrevocably lost; unlike traditional systems, there is no centralized copy of the cleartext data to recover. Therefore, the successful national-scale deployment of this approach must be coupled with robust, multi-factor key management, secure hardware modules, and a comprehensive disaster recovery protocol to mitigate these critical new single points of failure.

Looking ahead, our focus will shift to maximizing the efficiency and breadth of the system. We plan to dedicate substantial effort to optimizing encrypted arithmetic through methods such as batching, strategic parameter tuning, and integration with specialized hardware acceleration to enhance scalability and throughput. Furthermore, the system's utility will be extended to support a wider array of real-world scenarios, particularly complex deductions and credits that are currently challenging to model homomorphically. A key area of innovation involves embedding the CKKS homomorphic encryption scheme directly into Support Vector Machine (SVM) inference. By combining this with advanced feature engineering, we aim to increase the accuracy of SVM-based tax risk classification to 1.0 on encrypted data. Ultimately, this capability to provide secure audits, deliver scalable performance, and guarantee zero data exposure is poised to revolutionize global tax compliance, paving the way for governments, financial institutions, and service providers worldwide to modernize their operations. By doing so, our work not only fosters public trust and dramatically reduces breach risk but also accelerates the global transition to secure, robust digital taxation systems.

REFERENCES

[1] D. C. Snell, Ledgers and Prices: Early Mesopotamian Merchant Accounts. Yale Univ. Press, 1982.

[2] D. Patel, "Historical Evolution of Tax Laws: Key Developments — taxguru.in," https://taxguru.in/income-tax/ historical-evolution-tax-laws-key-developments.html, [Accessed 04-11-2025].

[3] _ adm virs, "A Global History of Taxation: From Ancient Tributes to Modern Systems — virsa.co," https://virsa.co/ a-global-history-of-taxation-from-ancient-tributes-to-modern-systems/, [Accessed 04-11-2025].

[4] K. McMahon, "Stressed About Taxes? Blame the Ancient Egyptians — smithsonianmag.com," https://www.smithsonianmag.com/history/ stressed-about-taxes-blame-the-ancient-egyptians-180984059/, [Accessed 04-11-2025].

[5] "History of Taxes: A Brief Overview — taxfoundation.org," https: //taxfoundation.org/taxedu/primers/primer-history-of-taxes/, [Accessed 04-11-2025].

[6] A. Kumar, "State tax policy from the oldest civilisation to Kautilya," International Journal of Science & Engineering Development Research, vol. 8, no. 2, pp. 443–445, Feb 2023, available online. [Online]. Available: http://www.ijrti.org/papers/IJRTI2302073.pdf

[7] "You have almost certainly been hacked — theweek.com," https: //theweek.com/articles/730439/have-almost-certainly-been-hacked, [Ac- cessed 04-11-2025].

[8] "The Ultimate Guide to Attack Surface — Netenrich — netenrich.com," https://netenrich.com/guides/attack-surface, [Accessed 04-11-2025].

[9] "Neiman Marcus says 64,000 affected by breach of Snowflake customer account — therecord.media," https://therecord.media/ neiman-marcus-snowflake-breach-thousands, [Accessed 04-11-2025].

[10] "Neiman Marcus confirms data breach after Snowflake account hack — bleepingcomputer.com," https://www.bleepingcomputer.com/news/security/neiman-marcus-confirms-data-breach-after-snowflake-account-hack/, [Accessed 04-11-2025].

[11] "Neiman Marcus Data Breach Litigation - Home — nmgsettlement.com," https://nmgsettlement.com/, [Accessed 04-11-2025].

[12] "IRS (2015-05-01) Cyber-Attack Hack Breach - The Cyber Security Incident Database (CSIDB) — csidb.net," https://www.csidb.net/csidb/ incidents/955f99e7-b45b-4d38-98a7-77379e2a749b/, [Accessed 04-11- 2025].

[13] "IRS: Crooks Stole Data on 100K Taxpayers Via 'Get Transcript' Feature — Krebs on Security — krebsonsecurity.com," https://krebsonsecurity.com/2015/05/ irs-crooks-stole-data-on-100k-taxpayers-via-get-transcript-feature/, [Accessed 04-11-2025].

[14] C. Gentry, "Fully homomorphic encryption using ideal lattices," in Proceedings of the forty-first annual ACM symposium on Theory of Computing, 2009, pp. 169–178.

[15] D. Trivedi, "Privacy-preserving security analytics," 5 2023. [Online]. Available: https://www.isaca.org/resources/news-and-trends/ isaca-now-blog/2023/privacy-preserving-security-analytics

[16] ——, "The future of cryptography: Performing computations on encrypted data," ISACA Journal, vol. 1, no. 2023, 2 2023. [Online]. Available: https://www.isaca.org/resources/isaca-journal/ issues/2023/volume-1/the-future-of-cryptography

[17] S. Angel, H. Chen, K. Laine, and S. Setty, "Pir with compressed queries and amortized query processing," in 2018 IEEE Symposium on Security and Privacy (SP). IEEE, 2018, pp. 962–979.

[18] J. W. Bos, W. Castryck, I. Iliashenko, and F. Vercauteren, "Privacy-friendly forecasting for the smart grid using homomorphic encryption and the group method of data handling," in Progress in Cryptology- AFRICACRYPT 2017: 9th International Conference on Cryptology in Africa, Dakar, Senegal, May 24-26, 2017, Proceedings. Springer, 2017, pp. 184–201.

[19] Boudguiga, O. Stan, H. Sedjelmaci, and S. Carpov, "Homomorphic encryption at work for private analysis of security logs." in ICISSP, 2020, pp. 515–523.

[20] F. Bourse, M. Minelli, M. Minihold, and P. Paillier, "Fast homo- morphic evaluation of deep discretized neural networks," in Advances in Cryptology–CRYPTO 2018: 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19–23, 2018, Proceedings, Part III 38. Springer, 2018, pp. 483–512.

[21] M. Kim and K. Lauter, "Private genome analysis through homomorphic encryption," in BMC medical informatics and decision making, vol. 15, no. 5. BioMed Central, 2015, pp. 1–12.

[22] D. Trama, P.-E. Clet, A. Boudguiga, and R. Sirdey, "Building blocks for LSTM homomorphic evaluation with tfhe," in International Symposium on Cyber Security, Cryptology, and Machine Learning. Springer, 2023, pp. 117–134.

[23] D. Trivedi, A. Boudguiga, and N. Triandopoulos, "Sigml: Supervised log anomaly with fully homomorphic encryption," in International Symposium on Cyber Security, Cryptology, and Machine Learning. Springer, 2023, pp. 372–388.

[24] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(leveled) fully homomorphic encryption without bootstrapping," ACM Transactions on Computation Theory (TOCT), vol. 6, no. 3, pp. 1–36, 2014.

[25] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in Advances in Cryptology– ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I 23. Springer, 2017, pp. 409–437.

[26] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," Cryptology ePrint Archive, 2012.

[27] C. Gentry, A. Sahai, and B. Waters, "Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based," in Advances in Cryptology–CRYPTO 2013: 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I. Springer, 2013, pp. 75–92.

[28] Chillotti, N. Gama, M. Georgieva, and M. Izabachene, "Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds," in Advances in Cryptology–ASIACRYPT 2016: 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part I 22. Springer, 2016, pp. 3–33.

[29] L. Ducas and D. Micciancio, "Fhew: bootstrapping homomorphic encryption in less than a second," in Advances in Cryptology– EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I 34. Springer, 2015, pp. 617–640.

[30] H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," Cryptology ePrint Archive, Report 2016/421, 2016, https://eprint.iacr.org/2016/421.

[31] Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," ACM Computing Surveys (Csur), vol. 51, no. 4, pp. 1–35, 2018.

[32] D. Boneh, A. Sahai, and B. Waters, "Functional encryption: Definitions and challenges," in Theory of Cryptography Conference. Springer, 2011, pp. 253–273.

[33] M. Abdalla, D. Catalano, R. Gay, and B. Ursu, "Inner-product functional encryption with fine-grained access control," Cryptology ePrint Archive, Paper 2020/577, 2020. [Online]. Available: https://eprint.iacr.org/2020/577

[34] U. Fiege, A. Fiat, and A. Shamir, "Zero knowledge proofs of identity," in Proceedings of the nineteenth annual ACM symposium on Theory of Computing, 1987, pp. 210–217.

[35] O. Goldreich, "Secure multiparty computation," Manuscript. Preliminary version, vol. 78, no. 110, pp. 1–108, 1998.

[36] "MPC Library — CoinFabrik — coinfabrik.com," https://www.coinfabrik.com/products/mpc-multi-party-computation-library/, [Accessed 10-11-2025].

[37] "Glossary — FHE, Differential Privacy & Multiparty Computation — dualitytech.com," https://dualitytech.com/glossary/, [Accessed 11-11- 2025].

[38] D. Bogdanov, L. Kamm, B. Kubo, R. Rebane, V. Sokk, and R. Talviste, "Students and taxes: a privacy-preserving study using secure computation," Proceedings on Privacy Enhancing Technologies, 2016.

[39] Burman, B. Johnson, V. L. Bryant, G. MacDonald, and R. McClelland, "Protecting privacy and expanding access in a modern administrative tax data system," National Tax Journal, vol. 77, no. 4, pp. 927–947, 2024.

[40] "Compliance FAQ — sarus.tech," https://www.sarus.tech/solutions/ use-cases/security-compliance/compliance-faq, [Accessed 10-11-2025].

[41] "Protecting Privacy: Differential Privacy and Homomorphic Encryption — The Central Texas IT Guy — thecentexitguy.com," https://thecentexitguy.com/protecting-privacy-differential-privacy-and-homomorphic-encryption/, [Accessed 10-11-2025].

[42] "Understanding Robust Privacy with Differential Privacy (DP) and Data Transformation Systems (DTS) - 7/25 - Azoo Blogs — cubig.ai," https://cubig.ai/blogs/ understanding-robust-privacy-with-differential-privacy-dp-and-data-\ transformation-systems-dts-7-25, [Accessed 10-11-2025].

[43] "Differentially private median and more — research.google," https://research.google/blog/differentially-private-median-and-more/, [Accessed 11-11-2025].

[44] C. Dwork, "Differential privacy," in International Colloquium on Automata, Languages, and Programming. Springer, 2006, pp. 1–12.

[45] ——, "Differential privacy: A survey of results," in Theory and Applications of Models of Computation: 5th International Conference, TAMC 2008, Xi'an, China, April 25-29, 2008. Proceedings 5. Springer, 2008, pp. 1–19.

[46] C. Dwork, A. Roth et al., "The algorithmic foundations of differential privacy," Foundations and Trends® in Theoretical Computer Science, vol. 9, no. 3–4, pp. 211–407, 2014.

[47] F. Barrientos, A. R. Williams, J. Snoke, and C. M. Bowen, "A feasibility study of differentially private summary statistics and regression analyses with evaluations on administrative and survey data," 2023. [Online]. Available: https://arxiv.org/abs/2110.12055

[48] J. Kato, E. O. Pinyi, I. D. Ssetimba, H. N. Nakayenga, B. Akashaba, and E. Twineamatsiko, "Securing taxpayer data: Advancing cybersecurity in tax accounting practices," International Journal of Research in Interdisciplinary Studies, vol. 2, no. 7, p. 42–46, Jul. 2024. [Online]. Available: https://journal.ijris.com/index.php/ijris/article/view/65

[49] D. Vangjeli, "Policy enforcement using attribute-based encryption in distributed environments," Master's thesis, Eindhoven University of Technology, August 2014, available at https://research.tue.nl/en/studentTheses/policy-enforcement-using-attribute-based-encryption-in-distribute/.

[50] Berke, T. South, R. Mahari, K. Larson, and A. Pentland, "zktax: A pragmatic way to support zero-knowledge tax disclosures," arXiv preprint arXiv:2311.13008, 2023.

[51] "What is a Zero-Knowledge Proof? — nmkr.io," https://www.nmkr.io/ glossary/zero-knowledge-proof, [Accessed 11-11-2025].

[52] "GitHub - sarojaerabelli/py-fhe: A Python library for fully homomorphic encryption — github.com," https://github.com/sarojaerabelli/py-fhe, [Accessed 10-11-2025].

[53] Ibarrondo and A. Viand, "Pyfhel: Python for homomorphic encryption libraries," in Proceedings of the 9th Workshop on Encrypted Computing & Applied Homomorphic Cryptography, 2021, pp. 11–16.

[54] T. Kluyver, B. Ragan-Kelley, F. Pe´rez, B. Granger, M. Bussonnier, J. Frederic, K. Kelley, J. Hamrick, J. Grout, S. Corlay, P. Ivanov, D. Avila, S. Abdalla, and C. Willing, "Jupyter notebooks – a publishing format for reproducible computational workflows," in Positioning and Power in Academic Publishing: Players, Agents and Agendas, F. Loizides and B. Schmidt, Eds. IOS Press, 2016, pp. 87–90.

[55] "Microsoft SEAL (release 4.0)," https://github.com/Microsoft/SEAL, Mar. 2022, Microsoft Research, Redmond, WA.

[56] "PALISADE Lattice Cryptography Library (release 1.11.2)," https:// palisade-crypto.org/, May 2021.

[57] Zama, "Concrete: TFHE Compiler that converts Python programs into FHE equivalent," 2022, https://github.com/zama-ai/concrete.

[58] Z. Brakerski, "Fully homomorphic encryption without modulus switching from classical gapsvp," in Proceedings of the 32Nd Annual Cryptology Conference on Advances in Cryptology — CRYPTO 2012 - Volume 7417. New York, NY, USA: Springer-Verlag New York, Inc., 2012, pp. 868–886, http://dx.doi.org/10.1007/978-3-642-32009-5\ 50.

[59] J. Fan and F. Vercauteren, "Somewhat practical fully homomorphic encryption," Cryptology ePrint Archive, Report 2012/144, 2012, https://eprint.iacr.org/2012/144.

[60] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "Fully homomorphic encryption without bootstrapping," Cryptology ePrint Archive, Paper 2011/277, 2011, https://eprint.iacr.org/2011/277.

[61] Benaissa, B. Retiat, B. Cebere, and A. E. Belfedhal, "Tenseal: A library for encrypted tensor operations using homomorphic encryption," 2021.

Received on 26-10-2025 Accepted on 24-11-2025 Published on 08-12-2025

https://doi.org/10.65879/3070-5789.2025.01.08