Pacific Northwest
NATIONAL LABORATORY

PNNL-36774

A Computational Review of Privacy-Preserving Mechanisms for the Smart Grid

September 2024

Javier E Ramirez
D. Jonathan Sebastian-Cardenas

U.S. DEPARTMENT OF ENERGY

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

PNNL-36774

A Computational Review of Privacy-Preserving Mechanisms for the Smart Grid

September 2024

Javier E Ramirez
D. Jonathan Sebastian-Cardenas

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Abstract

Smart grid technologies have rapidly become one of the largest and most comprehensive sources of data for the modern utility. For the most part, data streams are seen as an essential tool that enable utilities to carry out their day-to-day business operations, but they also create the need for efficient and secure data management strategies. In the context of the smart grid, ensuring data privacy is becoming an increasing concern due to a combination of factors that range from shifts in operational paradigms and rapid technology evolution to changes in legislation. Furthermore, researchers have highlighted the risks associated with improperly protected energy records. For example, energy consumption data from homes could be used to infer the behaviors and habits of home occupants through activity recognition or user profiling (Fan, 2017), which may lead to unfair service pricing, targeted advertising, or other personal security violations. Similarly, Electric Vehicles' (EVs) charging metadata could be used to reveal private information about the owner such as their payment methods, preferred charging stations, and other locational and timing information that could be used to reconstruct the vehicle owner's behaviors.

The privacy of user data, even when used for statistical analysis or machine learning training processes, also needs to be carefully considered, as an individual's private traits may still be vulnerable if their inclusion/exclusion greatly impacts the result or could be linked to a public dataset through cross-reference. The breach of user privacy also has severe impacts for organizations that store, transmit, or work on the data in the form of diminishing the public's trust in them while potentially incurring legal consequences (e.g., fines and suspensions under the European Union General Data Protection Regulation, Health Insurance Portability and Accountability Act, etc.). Because of these risks, several privacy-preserving mechanisms are available to help organizations comply with privacy legislations and prevent the unauthorized and malicious use of user data.

In light of these concerns, this report focuses on performing a computational review of privacy-preserving mechanisms that have received a significant amount of interest in literature. It specifically focuses on 1) homomorphic encryption, 2) zero-knowledge proofs, 3) differential privacy, and 4) federated learning. It is worth noting that although many of the methods presented in this document rely on cryptographic primitives, their intent is not to provide perfect secrecy, but rather to enable users to maintain privacy, and thus they shall not be compared or equated to other constructs that are aimed to address cybersecurity constructs.

Summary

As the grid moves away from traditional architectures to one that consists of distributed energy resources (DERs), smart meters, and other devices capable of transmitting and receiving data across the network, there is a growing need to ensure the privacy of the data. This data can contain information about a user's energy consumption/production, billing information, and other personal information. A malicious actor that obtains this data could use it for activities that range from learning customer behaviors (e.g., identifying vacation periods) to implementing unfair pricing or marketing strategies. From the grid perspective, if a malicious actor is able to access operational data that has not being adequately sanitized or privatized, malicious actors may use it to identify systematic weaknesses or vulnerabilities that, if exploited, could lead to operational consequences. While there are examples of how a malicious actor could cause damage to the grid (e.g. overloading transmission lines by tampering with the network topology to cause cascading outages and blackouts (Mousavian, 2013)), to the authors' knowledge, there has not being an example of the grid system being exploited that leads to operational consequences as a direct result of privacy violations. This however does not imply that such events do not create (or will not create) impacts. For the utility provider, the leakage of user data can have severe consequences in the form of fines and other penalties for violating local laws and regulations (e.g., European Union General Data Protection Regulation (GDPR)). A data breach would also erode the public's trust in them, and other organizations may be less inclined to collaborate with them.

Various privacy-preserving mechanisms can be applied to existing frameworks to protect the data of the user and the grid. Homomorphic encryption (HE) is a form of encryption that allows ciphertexts, obtained from applying mathematical algorithms and an encryption key to plaintext data, to be worked on without the need to first decrypt them back into plaintext. This greatly reduces the number of situations where a plaintext can be viewed in the clear and enables collaboration with entities that would otherwise not have access to the data due to privacy concerns. Its main drawback is that it is more computationally expensive than other privacy-preserving technologies, especially in the case of Fully Homomorphic Encryption (FHE) that can perform an unlimited number of operations for an unlimited number of applications at the cost of requiring a relatively expensive bootstrapping process that re-encrypts the ciphertext (Acar, 2018). Zero-knowledge proofs (ZKP) grant one entity (i.e. the prover) the ability to prove that they know a secret to another entity (i.e. the verifier) without leaking any information other than the fact that they know the secret (Chen, 2023). The use of ZKPs allow entities to remain anonymous during authentication processes, and the overhead is relatively low when compared to other public key infrastructures. One of its drawbacks is the tradeoff between efficiency and privacy when using a trusted third party in the authentication process (Sun, 2021). Differential privacy (DP) is an approach to protecting privacy when statistical analysis is performed on a dataset containing data from a group of individuals (Dwork, Differential privacy, 2006). It provides privacy guarantees that the inclusion or exclusion of an individual's dataset in the analysis will not affect the results greatly enough for an adversary to gain any meaningful information about them. Its primary drawback is that it cannot provide complete privacy preservation on its own and requires the use of other privacy-preserving mechanisms (Husnoo, 2021). Federated learning (FL) is a method of training a machine learning global model that doesn't require the clients to upload their raw data to a central server. The server instead sends a global model to clients, who then train the model on their local raw data and upload the updated model parameters to the server for aggregation. The heterogeneity of the devices on the network and the costs of communication are some of the drawbacks to FL (Banabilah, 2022). There is also no standardized method of benchmarking in the field of FL, which makes comparing various algorithms difficult (Almanifi, 2023). This paper provides more details on these privacy-preserving technologies and their benefits, applications, challenges, and computational overhead.

Acknowledgments

This project was supported by the Department of Energy, Office of Electricity, Advanced Grid Research and Development Program. The authors would like to thank Chris Irwin for his support and contributions to shaping the scope and direction of this work.

Acronyms and Abbreviations

BFV – Brakerski-Fan-Vercauteren
BGN – Boneh-Goh-Nissim
BGV – Brakerski-Gentry-Vaikuntanathan
CIFAR10 – Canadian Institute for Advanced Research
CKKS – Cheon-Kim-Kim-Song
CPA – Chosen Plaintext Attack
DER – Distributed Energy Resource
DLP – Discrete Logarithm Problem
DP – Differential Privacy
EV – Electric Vehicle
FHE – Fully Homomorphic Encryption
FHEW – Fastest Homomorphic Encryption in the West
FL – Federated Learning
FTL – Federated Transfer Learning
GDPR – General Data Protection Regulation
HE – Homomorphic Encryption
HEAAN – Homomorphic Encryption for Arithmetic of Approximate Numbers
He-Lib – Homomorphic-Encryption Library
HFL – Horizontal Federated Learning
HPRE – Homomorphic Proxy Reencryption
IND-CCA – Indistinguishability Under Chosen Ciphertext Attack
IoT – Internet of Things
ISO – Independent System Operator
ML – Machine Learning
MNIST – Modified National Institute of Standards and Technology
MPC – Secure Multiparty Computation
NZKP – Non-interactive Zero-Knowledge Proof
PCP – Probabilistically Checkable Proofs
PET – Privacy Enhancing Technology
PHE – Partially Homomorphic Encryption
PKI – Public Key Infrastructure
PRE – Proxy Reencryption
QAP – Quadratic Arithmetic Programs
R-LWE – Ring-Learning With Error
RSA – Rivest-Shamir-Adleman
RTO – Regional Transmission Organization

SCADA – Supervisory Control and Data Acquisition
SCI – Scalable Computational Integrity
SEAL – Simple Encrypted Arithmetic Library
SWHE – Somewhat Homomorphic Encryption
TFHE – Fast Fully Homomorphic Encryption
VFL – Vertical Federated Learning
ZKP – Zero-Knowledge Proof
ZKRP – Zero-Knowledge Range Proof
Zk-SNARK – Zero-Knowledge Succinct Non-Interactive Argument of Knowledge

Contents

Figures

Tables

1.0 Introduction

The smart grid is often seen as a necessary component for building a more efficient, reliable, and sustainable grid (Kumar, 2019). In its simplest form, the smart grid can be defined as a network of highly interconnected assets that constantly communicate with each other in order to satisfy a wide variety of operational goals. Despite its relatively simple definition, the smart grid is a complex system of systems that must be broken into subcomponents before it becomes feasible for engineers and operators to understand and manage its inner workings. Based on this premise, the smart grid is often divided into subdomains that align with other existing regulatory and organizational structures that are present in the energy industry. For simplicity, this work adopts the NIST domain classification, which can be summarized as follows:

- The function of the bulk generation domain is to generate electricity for the consumers (Dileep, 2020). It is electrically connected with the transmission domain and has strong data interdependencies with the transmission, markets, and operations domains through multiple shared interfaces. Traditional information flows focus on acquiring operational information that can be used to assess system performance and to ensure any potential issues are rapidly addressed (e.g., generator failure).
- The transmission domain transfers the electricity generated from bulk generation units to the distribution system via a network of transmission lines and substations. An independent system operator (ISO) or regional transmission organization (RTO) is responsible for balancing supply and demand over the network. Typical examples of operational data collected include data from phasor measurement units and state variables obtained via Supervisory Control and Data Acquisition (SCADA) systems. These data streams may be complemented with economic indicators under the market domain.
- The distribution domain is responsible for taking the electrical energy obtained from the transmission domain and allocating it to the consumers/prosumers. It consists of objects such as capacitor banks, storage devices, and sectionalizers. It coordinates with the operations domain to manage the power flows or with the market domain to signal prices.
- The consumers/prosumers domain consists of customers who can control their energy consumption and generation. Traditionally, this domain mostly interacts with the services domain via smart meters or local controllers. However, as technologies such as Inverter Based Resources , Electric Vehicles (EVs) , among other Distributed Energy Resources (DERs) continue to emerge, advanced energy services interfaces may enable this domain to more actively interact with other domains.
- The services domain is usually tasked with billing and account management. The services domain communicates with the operations domain to maintain system control and situational awareness, and the consumers/prosumers and markets domains for economic growth through the creation of new products and services. It is envisioned to eventually support the emergence of advanced grid services across all domains.
- The operations domain typically includes supervisory agents and other regulatory stakeholders that are responsible for maintaining optimal operation of the power system.

- The markets domain consists of actors that balance supply and demand using economic signals. Due to their importance, a series of regulatory protections are often added to ensure all processes remain fair and efficient while mitigating risks.

Traditionally, attention was set on the generation, transmission, and distribution domains; however, as more DERs continue to be added, the consumer/prosumer domain is expected to increase its importance. In addition, the service domain eventually may play a more active role, supporting the emergence of decentralized grid services, which may enable a greater degree of system resiliency while also empowering the consumer/prosumer domain to more actively engage in satisfying grid objectives. Such capabilities are expected to build upon the communication capabilities of the smart grid and will almost certainly leverage information exchange as the main mechanism to achieve multi-agent coordination.

Although the grid of the future may still be under development, the current generation grid is already experiencing high levels of interdependency, requiring a constant exchange of information to function appropriately. Traditionally, such exchanges involved a limited set of actors, i.e., a local utility company exchanging data with an RTO/ISO. In addition to the limited number of actors, these data exchanges are often subject to strict regulatory and organizational policies that dictate exchanges based on business needs, conferring a certain level of inherent protections against potential data misuse. However, as consumers continue to play a more active role in grid operations (e.g., via DERs) and new service providers continue to emerge (e.g., DER aggregators), it has become apparent that existing data management practices may not provide the necessary privacy protections to ensure all system participants operate on equal ground, and, specifically, the topic of privacy has been identified as a key enabler.

The concept of privacy is by itself is a complex, multi-disciplinary term that has no universally agreed-upon definition, but it can generally be defined *as "the ability of individuals to control the terms under which their personal information is acquired and used"* (Culnan & Bies, 2003). Despite the lack of a formal definition, for the purposes of this work, the term privacy will be used to refer to digital privacy, a subset of information privacy that *"focuses on the proper handling and usage of sensitive data that are generated and transmitted within digital environments"*[1] (IEEE digital privacy group, 2023). At a high level, the successful adoption of privacy features in the smart grid is expected to:

- Encourage innovation while appropriately protecting the privacy and confidentiality of all participants. Thereby enabling access to reliable and sustainable energy while also addressing societal aspects.

- Provide participants with the ability to manage their own data. This may include determining the *"Who, What, When, Where, and Why"* in the context of their data (Sebastian Cardenas, Mukherjee, & Ramirez, 2023).

Clearly, privacy is a desirable system property that, if well implemented, could be used as a tool to enable the grid of the future. However, privacy is a complex topic that requires a multi-disciplinary approach for its implementation to be successful. Within the context of this report, focus will be placed on analyzing the computational tools (i.e., Privacy Enhancing Technologies) that could be used to address the privacy requirements of the smart grid. As illustrated by Figure 1, Privacy Enhancing Technologies represent one of the foundational constructs that can

[1] Although the citation is verbatim, "the proper handling and usage of sensitive data" does not have a universal answer, because the answer depends on the domain, the stakeholder preferences and other local regulations and perspectives (which themselves shift with time).

enable the emergence of secure, scalable, and flexible smart grid deployments by providing the necessary technical constructs that allow application designers to build and maintain privacy-aware solutions.

Privacy Enhancing Technologies (PETs) is a broad term used to encompass a series of computational and mathematical methods that have been engineered to address specific privacy threats (e.g., to prevent dataset disaggregation) or to enable a new capability (e.g., to prove an attribute without revealing the original data set). Due to their specialized nature, each PET must be carefully selected based on a combination of application-specific needs and PET-specific characteristics to maximize compatibility and to ensure that the resulting data artifacts meet the necessary privacy needs.

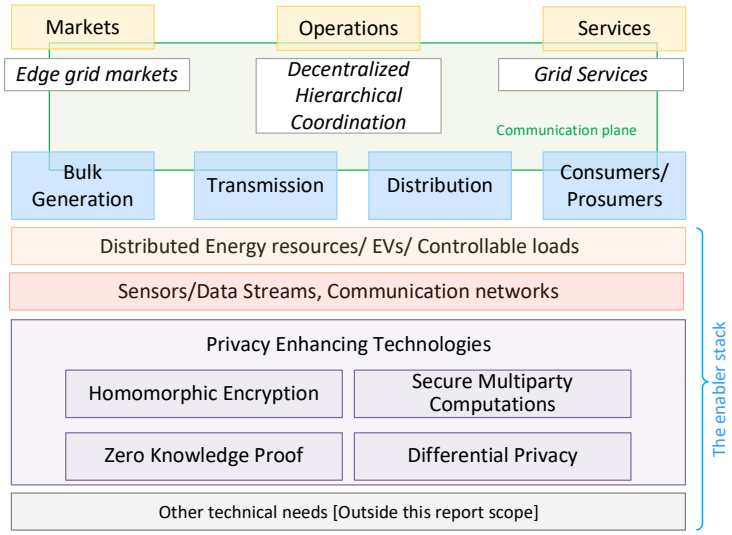


Figure 1: Privacy Enhancing Technologies as an enabler for the modern smart grid.

As discussed above, each privacy technology introduces its own set of requirements and potential drawbacks that must be considered before they are integrated into a smart grid application. In alignment with this need, this report seeks to present a technical review of various PETs and offer application engineers the necessary guidance for a successful application deployment. In the future, this report is expected to be complemented by follow-up guides that will continue to support decision makers in the deployment of secure, scalable, and reliable privacy-aware applications.

1.1 Privacy in the Traditional Power Grid

Under the traditional operational paradigm, system reliability was largely supported by a predictable demand and supply growth that enabled system planners to develop and schedule infrastructure upgrades before any reliability issues became apparent (Currie, 2023). However, with the wide-spread adoption of DERs and the ongoing electrification process that is currently happening across many other economic sectors (e.g., EVs in the transportation domain), there is a growing need for methods that can react to the rapid changes that arise from the variability and ad-hoc nature of such systems. In this context, remote sensing technologies, such as the smart meter, are seen as key enablers for a more responsive grid (Dileep, 2020). Although primarily used for billing purposes, smart meters can be leveraged to fulfill a wide variety of tasks that range from supporting improved grid observability to enabling advanced pricing strategies that can be used for dynamic grid control. From a utility perspective, their *"on-*

demand" communication capabilities can provide multiple operational benefits (e.g., rapid outage detection and accelerated restoration), but they can also be used to offer improved customer experiences that go beyond the grid's operational objectives. This may include providing access to granular consumption records and other advanced analytics that enable customers to optimize their energy use. However, these capabilities have also raised privacy concerns due to the amount of personal information that may be extracted or inferred outside of their intended usage applications (e.g., behavioral patterns such as household occupancy and appliance usage, etc.) (Currie, 2023). Such risks could further increase if records are made publicly available (e.g., due to system compromise), which may enable malicious agents to infer information that is potentially harmful to the household.

In accordance with these risks, multiple researchers have proposed a series of mechanisms and tools that could be used to limit the potential risks associated with the collection, transport, and storage of grid data. For example, in (Finster, 2015), the researchers propose to address the problem of securing data privacy in the smart grid by first classifying its end use as either an authoritative or informative data source and then applying appropriate mitigations. An authoritative role may be relevant when the information is used to produce an accurate billing statement, while an informative role may be relevant when the meter data is used to complement or augment an existing data stream (e.g., to improve grid visibility). Under such a classification system, billing-like applications could implement a trusted third party that oversees the behavior of customers, with each customer being responsible for computing their own bill (e.g., via trusted computing or cryptographic functions) and then reporting the net cost back to the utility without revealing unnecessary data. In (Asghar, 2017), the researchers provided an overview of the challenges, recommendations, and research directions for preserving the privacy of smart meter data during its transmission and management phases. The researchers focused on both a trusted operator model (where the operator collects and stores the data) and a non-trusted operator model (where all entities are untrusted, and the data must be manipulated in some way to preserve privacy). The authors identified a series of systematic issues that ranged from the user's inability to verify data usage according to the agreed consent, to risks inherent to the mechanisms used to audit energy consumption.

In (Abdalzaher, 2022), the researchers described vulnerabilities in smart meters and evaluated eleven trust models designed to preserve data privacy, while also providing recommendations on which models should be used based on their evaluation. The models included game theory, clustering, Bayesian, entropy, fuzzy logic, differential privacy, machine learning*, Kullback-Leibler* Divergence, generative adversarial privacy, data aggregation and pseudonyms. A further, high-level overview of techniques that have been used in literature to enhance the privacy attributes of the smart grid can be observed in Table 1.

Table 1: Privacy Enabling Technology Applications to Smart Grid

Citation	PETs used	Overall application or problem being solved.
(Taïk, 2020)	Federated Learning (FL)	A short-term demand forecasting scheme was assembled based on FL. The proposed scheme was tested using demand data from 200 homes that featured similar characteristics. Four scenarios were used to evaluate the accuracy vs privacy benefits of FL, this was done by changing the number of homes used to train the global model [5,20] and varying number of rounds used for local training [1,5]. The results found that model accuracy is dependent on household characteristics, and that pre-clustering and post-retraining (local) could improve accuracy.
(Badr, 2023)	Federated Learning, Inner-	The researchers developed a FL-based energy prediction system aimed at achieving high accuracy while preserving the customer's

	Product Functional Encryption	private information. Inner-product functional encryption (IPFE) was used on the parameters of the customer's model so that the data could be sent to the utility provider anonymously. The scheme used by the researchers enabled the utility provider to use the encrypted parameters to build a global model.
(Wang, et al., 2022)	Elliptic-curve Cryptography, Digital Watermark	The researchers presented a method for hiding sensitive smart meter data (e.g., user ID, location) using elliptic-curve cryptography (ECC) and digital watermarks. Sensitive information is first encrypted with ECC. Measurements appear as regular floating-point numbers (encoded in base64), but the mantissa contains a watermark. The smart meter determines the content and order in which the watermark is to be embedded. Once encoded, the modified data is repacked into a new sequence to be transmitted to the control center. Order is randomly generated based on UNIX time to prevent watermark manipulation, preserving privacy.
(Streppara va, 2022)	Homomorphic Encryption (HE)	The researchers demonstrated that anonymous aggregation of energy production and consumption data was possible through a cryptographic protocol based on HE. The protocol was deployed on an add-on computing module attached to the meter's optical port. The test results showed that the computational burdens were acceptable for encoding, but due to their cost, decoding should be reserved for auditing purposes. The researchers suggested the use of zero-knowledge proofs when access to exact values were needed (e.g., for billing purposes).
(Syed, 2020)	Partially Homomorphic Encryption (PHE), Fully Homomorphic Encryption (FHE)	The use of PHE and FHE was used to train classical and deep learning Artificial Intelligence (AI) models. The classical model performed linear regression for load forecasting purposes by leveraging PHE to protect the training data. The load forecasting application showed that the Root Mean Square Error (RMSE) was 0.0352 MWh under their framework (vs 0.0248 MWh without encryption). The deep learning model was used to create a fault classification and localization application using FHE to protect the underlying data. Tests performed by the researchers determined that the AI model was able to perform fault localization with an accuracy of 97-98% under their framework (vs 98-99% when the model was trained on plaintext data).
(Xu, 2023)	Fully Homomorphic Encryption (FHE)	A privacy-preserving framework using the Lattigo FHE library was proposed and evaluated by researchers to determine its ability to compute statistical measures in residential energy. Computations intended to represent a household were simulated with a MacBook Pro with a CPU speed of 2.3 Ghz and 4GB of DDR3 RAM. Computations calculated by all other parties were done on a Windows laptop with a processor speed of 1.90 Ghz and 16 GB of RAM. With an encryption key length of 128 bits, the researchers found that summation took 58,235 ms and variance calculation took 127,423 ms to process 100 households.
(Pop, 2020)	Zero-Knowledge Proof (ZKP), Blockchain, Distributed Ledger Technology (DLT)	A decentralized implementation of demand response program based on blockchain and ZKPs was presented by the researchers. ZKPs are utilized to obscure the prosumer's energy data and requested flexibility profiles. The blockchain stores the deviation quantities and performs ZKP validation to verify the deviation is correctly computed. According to the researchers, the scheme effectively safeguarded the integrity and confidentiality of the prosumers while also improving a demand response programs audit and financial settlement capabilities.

(Hassan, 2019)	Differential Privacy (DP)	A DP-based real time load monitoring approach was used to preserve the privacy of the user's routines and their renewable energy usage. The scheme aggregates the net demand for a given timeframe (denoted as S). If S is greater than 0, then: $$SN = S + N + EX$$ Where N is the Laplacian noise being added, and EX is an offset value. SN is compared with a pre-defined maximum peak load value P, and if SN is greater than P, the value of SN is set to P and transmitted to the utility. The value of $SN - P$ is stored in EX, which is the offset for the next timeframe. Their scheme was evaluated using the Residential Energy Consumption Survey for the Midwest, and the solar and wind energy dataset from the Hong Kong Observatory. The data was iterated over 31 days at a 10-minute interval. The monthly results showed that their scheme achieved an error rate of only 1.5%. (When P was 1.2 kWh, the total energy reported was 4.84 kWh and the masked value was 4.91 kWh).
(Tran, 2022)	Differential Privacy (DP) through noise generation and distribution	The researchers demonstrated a privacy system whose goal was to anonymize electricity consumption collected by smart meters. The proposed mechanism seeks to prevent attackers from inferring the resident's habits and home appliances used. The system used two separate algorithms to generate noise at the smart meter and at the Distribution System Operator (DSO) endpoint. The noise is generated using a private noise distribution protocol called nn-PND, which is a noisy neural network model that seeks to maintains data utility while preserving privacy.

Based on the research presented in Table 1 it is clear that a wide variety of PETs have been proposed to address the data privacy challenges introduced by the adoption of smart grid technologies. Despite its potential advantages, some of the proposed PETs may require significant changes to the existing operational paradigm before they can be successfully adopted by the industry, while others may be infeasible to adopt in their original form due to scalability issues or other performance bottlenecks. Hence it is imperative that potential adoption barriers are identified and addressed to further accelerate the deployment of privacy-aware grid solutions. In alignment with this need, this paper focuses on analyzing the computational overheads of some of the most employed privacy-preserving mechanisms found in literature. In particular, this report focuses on: A) homomorphic encryption (Section 2.0); B) zero-knowledge proof (Section 3.0); C) differential privacy (Section 4.0); and D) federated learning (Section 5.0). Noise-based digital watermarking was initially considered in the scope of this report, but the effort was abandoned due to the limited amount of research detailing its computational complexity and a lack of security guarantees that could lead to potential abuse (See Appendix A). Each section of this report introduces the privacy-preserving mechanism, followed by its benefits, applications, challenges, and computational overheads (if applicable). Finally, Appendix B provides additional information on the notation used to express time and space complexities, as well as simple numerical examples of each technology being covered.

As stated earlier, this report seeks to provide a technical analysis of the computational overheads involved in the adoption of PETs in the context of smart grid-related applications. Specifically, this document aims to present a series of quantitative and qualitative comparisons that enable application engineers to identify a method's strengths, risks, and potential limitations. These comparisons are aimed to foster a questioning attitude that promotes fair comparisons between different technology approaches and ensures that solutions remain secure, scalable, and efficient in the long term.

2.0 Homomorphic Encryption

Modern encryption represents a set of methods and mechanisms used to provide data confidentiality, ensuring data can only be accessed by parties who hold the appropriate credentials. Although privacy and confidentiality are interrelated, they are not necessarily equal, yet maintaining a certain level of confidentiality is one of the key requirements towards achieving a comprehensive privacy solution. At its core, encryption ensures data confidentiality by transforming human-readable data (called the plaintext) into a random string of characters (called the ciphertext). An encryption algorithm and an encryption key (which is a random string of characters itself) is applied to the plaintext to transform the input and generate the ciphertext. To reverse the process (i.e., to recover the plaintext) a decryption algorithm and a decryption key are applied to the ciphertext. One of the limitations with traditional encryption methods, however, is that computations (e.g., mathematical operations) can only be done over the plaintext, requiring all parties involved in such computation to have access to the encryption key (Alharbi, 2020). This introduces opportunities for malicious actors to exfiltrate or abuse the data being shared, or to misuse the decryption keys (e.g., sharing a key with unauthorized users). These problems tend to grow as the number of involved parties expand or as conflicting role dynamics arise. For example, if there are n number of entities that need to work on the data, then there are n locations that may leak or abuse the underlying data streams. Another common concern is that parties may continue to access past and future data unless the key is changed and the previously encrypted data is destroyed (Acar, 2018).

To address these challenges, Homomorphic Encryption (HE) has introduced a set of features that enable computational operations to be performed without requiring an implicit decryption step (Acar, 2018). By leveraging HE, data-producing agents can freely share data in its encrypted form with entities that can apply functions and transformations without any key sharing, theoretically eliminating the risks associated with untrusted third-party processors. Since the cleartext is only available at the encryption or decryption stages, the only relevant threat entry points are the source and receiver system. At the time of writing, HE is still an emerging research topic, and hence some limitations exist in the number and types of operations available due to a combination of scalability and performance limitations. Many common operations such as addition, subtraction, multiplication, and division (i.e., by multiplying by 1/scalar) are already well-understood and are available to application developers. However, it is worth providing a quick classification of different HE methods according to their processing capabilities. These are:

- Partially homomorphic (PHE): These methods enable users to perform a single operation or family of operations infinitely. For example, a method may support an infinite number of cumulative additions (and subtractions), but without multiplication support, or vice versa.

- Somewhat homomorphic (SWHE): Multiple operations can be applied a limited number of times. For example, a SWHE algorithm may offer the ability to support an infinite[1] number of additions followed by a finite number of multiplications.

[1] Some SWHE algorithms introduce a non-trivial amount of noise while performing operations, potentially limiting the number of same-type operations and combinations thereof (which may also impact the order that these can be applied). Further details will be presented during the bootstrapping discussion.

- Fully Homomorphic (FHE): Any operation can be applied infinitely. The types and number of operations are unbounded, but practical limits may exist due to computational overheads or impractical message lengths.

Some examples of PHE schemes that are in use today include *Rivest–Shamir–Adlema* (RSA), *El-Gamal*[1], and *Paillier*. RSA is a multiplicative PHE scheme that was introduced in (Rivest, 1978). It has been applied to banking and credit card transactions (Doan, 2023), where it is used in the transport layer security (TLS) protocol to encrypt sensitive data and for authentication (Soram, 2015). An example of this scheme is detailed in Appendix B.2. The security of the scheme relies on the difficulty of factoring the product of two large prime numbers. *El-Gamal* is another multiplicative PHE scheme that was introduced in (ElGamal, 1985) that provided improvements over the traditional Diffie-Hellman Key Exchange algorithm (Diffie, 1976). The security of *El-Gamal* relies on the difficulty of solving discrete logarithms. *Paillier* is an additive PHE that was first described in (Paillier, 1999), which has been used in e-voting mechanisms. Its security is based on the composite residuosity problem (Doan, 2023).

Despite the limited number of applications in which PHE capabilities are directly applicable, their theoretical potential rapidly became a driver for developing powerful constructs that could satisfy the needs of real-world applications. This initiated a wave of research that resulted in a first generation of SWHE algorithms that included algorithms such as the *Polly Cracker* scheme, which enabled additive and multiplicative operations to be carried on ciphertexts. Unfortunately, its main drawback was that the ciphertext's size grew exponentially which in turn led to increased computational times, resulting in poor scalability (Fellows, 1994). Subsequent SWHE schemes sought to address some of these performance drawbacks, leading to schemes such as the *Boneh-Goh-Nissim* (BGN) which was introduced in (Boneh, 2005). BGN's security model is based on the subgroup decision problem and allows an unlimited number of additive operations and a single multiplication over ciphertexts while maintaining the size of the ciphertext as a constant. Despite these advancements, schemes such as BGN lacked a sufficient level of generalization to be applicable in real-world scenarios.

Driven by these limitations, the first feasible implementation of a FHE scheme was proposed as a reusable blueprint in (Craig, 2009). The blueprint relies on the iterative application of a two-stage process, referred to *squashing* and *bootstrapping*. The first stage seeks to create a simplified representation of the ciphertext (referred to as squashing) by adding auxiliary information to the ciphertext, helping to reduce the computational complexity associated with performing an evaluation. Bootstrapping performs a "refresh" that removes or "resets" the noise introduced during the computation process by essentially applying another encryption function over the old ciphertext (without decrypting it first), enabling FHEs to achieve an unlimited number of operations over the ciphertext. The bootstrapping process is the main drawback of FHE, since it adds a noticeably large amount of computational cost. Some commonly used FHE schemes are BFV (*Brakerski, Fan, and Vercauteren*), BGV (*Brakerski, Gentry, and Vaikuntanathan*), and CKKS (*Cheon, Kim, Kim, and Song*). CKKS is a notable FHE scheme because it performs computations on ciphertexts containing real or complex values (Doan, 2023). The security of many FHE schemes is based on the *Ring-Learning With Errors* (R-LWE) assumption (Podschwadt, 2022). As expected, HE methods' strengths rely on their ability to balance confidentiality with the ability to efficiently carry out operations, hence the majority of these schemes must be described and proven using strong mathematical language, a feature that can limit their target audience. To help bridge the gap between theory and practice, works

[1] *El-Gamal* is also referred as *ElGamal and El Gamal*, this work has adopted the former to ensure consistency.

such as (Acar, 2018) may serve as an initial reference to application developers seeking to familiarize themselves with a method's principle of operation, operational assumptions, and typical use cases.

2.1 Benefits and Applications of Homomorphic Encryption

A major benefit of using HE is that it reduces the number of instances in which the data can be viewed in plaintext, thus improving its security. By maintaining the ciphertext in an encrypted form, a large number of computational operations can be performed while ensuring the plaintext is never exposed. This property of HE is especially useful in protecting user data privacy in cases where large amounts of data must be processed by untrusted parties, such as in machine learning applications, where a model's accuracy may be dependent on the availability to access large amounts of data (Marcolla, 2022).

By offering data processors the ability to compute functions without having raw data access, HE may assist business and organizations in complying with laws and policies regarding data privacy (e.g., EU General Data Protection Regulation, California Consumer Privacy Act, etc.). While violations of these laws can result in severe fines and other penalties, there are secondary impacts to an organization, such as diminished trust and negative publicity, potentially resulting in fewer businesses or clients willing to collaborate with them. HE could also enable collaboration between mutually-distrusting parties by providing each party with strong guarantees that assert their data cannot be extracted or re-shared in a meaningful manner with third parties. Due to these characteristics, HE has been proposed in multiple applications, across the domains of energy, healthcare, machine learning, finance, smart homes, and more. Table 2 lists some real-world applications of HE.

Table 2: Applications of HE Across Various Domains

Citation	Keywords	Overview
(Yuan, 2024)	Electronic Voting, Paillier, Partial Knowledge Proof, Timed-Release Encryption	The researchers proposed a timed-release e-voting system that implemented Paillier PHE and partial knowledge proofs. Paillier was used to maintain confidentiality while tallying votes and to ensure plaintext information could not be accessed before a previously agreed-upon time. Partial knowledge proofs were used to preserve the privacy of the ballot's content while still verifying its legitimacy. The researchers also conducted a security and performance analysis on their scheme. The security analysis determined that an eavesdropper could not break the semantic security of the time-delayed HE scheme in order to decipher ciphertexts. The performance analysis showed that their scheme could be applied to practical e-voting scenarios.
(Vengadapurvaja, 2017)	Healthcare, Medical Images, Cloud Computing, FHE	The researchers proposed a method based on HE to encrypt and perform computations on medical images to preserve the patient's confidentiality. The method consists of transforming the image into a matrix and sending the separate elements to the HE algorithm to generate public and private keys. The keys are then used for encrypting the matrix elements that can be stored in a public cloud. When needed, the matrix elements are sent to the decryption algorithm and then converted back into the original image for medical diagnosis. Analysis of the key space (2^{150}) determined that the proposed method was resistant to brute force attacks.
(Lee J.-W. H., 2022)	Privacy-Preserving Machine	The researchers implemented a standard ResNet-20 model (an image recognition engine) via RNS-CKKS FHE and applied approximation methods and bootstrapping techniques to improve

	Learning, *(Residue Number System)*, RNS-CKKS FHE Scheme	the accuracy of the model while preserving data privacy. The approximation methods were used to evaluate non-arithmetic functions with adequate precision. The bootstrapping technique was utilized to allow the evaluation of *"an arbitrary deep learning model on encrypted data."* The proposed model was tested with the CIFAR-10 dataset (an image training repository) and found that it accurately replicated 98.43% of the results produced by a non-privacy-aware ResNet-20 model.
(Yu, 2021)	Road Distance, Road Network	In order to preserve the privacy of a user's location data when performing road distance computations, two HE schemes were built and evaluated by the researchers. The first approach relied on the Paillier algorithm, while a second approach based on the *Fan-Vercauteren* (FV) scheme (a SWHE-based scheme), which was combined with a custom road model. The two schemes were evaluated on a city's road network which consists of 21,048 vertices and 21,693 edges. The results of the simulations found that the proposed solutions were able to compute the road distance with an 88-95% accuracy without requiring accurate location information.
(Yucel, 2019)	Electric Vehicle, Vehicular network	Researchers used the Paillier PHE and the *Bichromatic Mutual Nearest Neighbor* (BMNN) algorithms to provide customers with locational privacy in vehicle-to-vehicle (V2V) applications. To demonstrate the proposed method, a privacy-aware matching engine that allows users to find potential matches in V2V charging applications was developed. The proposed matching algorithm was simulated with 100 requestors and 100 suppliers operating over a $1km^2$ area. The results determined that there were lower average waiting times and less overhead when compared with a classical, centralized matching approach.
(Cheng Z. F.-Y., 2021)	Energy management system, optimal power flow	A private, collaborative distributed energy management system (P-CoDEMS) was developed and evaluated by the researchers with the goal of preserving user privacy when solving an AC optimal power flow (AC-OPF) problem. P-CoDEMS utilized a custom optimization technique and the *Homomorphic Encryption for Arithmetic of Approximate Numbers* (HEAAN) scheme. It was tested on four representative distribution systems (22-bus, 69-bus, 85-bus, and 141-bus); The results showed that their proposed system computes the AC-OPF while preserving user privacy.
(Sambasivarao, 2024)	Secure Energy Trading, GreenTrade Platform, Blockchain, DLT, Simulation Framework	The researchers proposed an energy trading framework using HE to preserve the privacy of energy trade data and increase the security and efficiency of energy trading on the GreenTrade platform. The framework was tested on data simulated via Python. The simulation involved randomizing energy needs, supply, and prices to mimic realistic transactions. The simulations showed that the addition of HE produced minimal computational overhead. However, the researchers noted that their evaluation focused on a limited set of performance metrics, and did not account for real-world conditions (e.g., network latency, communication overhead).
(Lei, 2022)	Smart Grid, Smart Meter	The researchers implemented a privacy-aware, energy trading platform. Hyperledger Fabric channel-isolation capabilities were leveraged to fragment and isolate network members. Paillier was used to encrypt, and thus protect private data (e.g., account balance). Multiple performance tests were carried out to determine timing overheads. When using a key length of 4,096 bits, encryption took about a second on a single core machine. When a batch encryption was performed over 1000 records, the operation took approximately 82 seconds. On a separate test, a 16-core system

		completed the same operations in 5.3 seconds. Since production systems will likely be multi-core, the researchers determined the added computational cost of Paillier is an acceptable tradeoff.

2.2 Challenges and Attacks Against Homomorphic Encryption

The biggest issue with FHE schemes is that squashing and bootstrapping procedures are required to transform a SWHE scheme into a FHE scheme, which significantly increases the computational and storage costs of adopting these methods (Acar, 2018). These processes are needed because additive and multiplicative operations add "noise" to the ciphertext, which is a needed feature to guarantee that the ciphertext maintains its intended levels of security. Additive operations increase the "noise" linearly, while multiplicative operations increase it exponentially, imposing limits on the overall number of operations that can be carried out. Once a certain threshold is reached, the ciphertext can no longer be successfully decrypted. As mentioned earlier, squashing is the process of reducing a decryption algorithm's complexity, a process that must be tied to the bootstrap phase (Craig, 2009). The bootstrapping process involves re-encrypting the bootstrappable ciphertext to obtain a new ciphertext and thus reset the noise level. Although the concept of bootstrapping is a relatively simple operation to understand, implementing an efficient, functional algorithm remains a subject of active research that has resulted in significant time improvements (e.g., from hours to seconds) and storage reductions (from GB to KB) over the last decade.

On the other hand, PHE-based methods are usually efficient. However, due to only supporting one type of operation (which may be applied infinitely, in some cases), PHE schemes present an adoption challenge where a combination of multiple or complex computations are required (Awadallah, 2020). This property may force application developers to manually string multiple PHE schemes before they can be adopted in their application (if at all possible). Another issue facing PHE schemes is that large key and message sizes can introduce processing delays that render the system unusable.

Despite the benefits of HE schemes, certain applications may not be aligned with the features provided by HE due to scalability concerns (e.g., excessive computation delays) or mismatches between the application's needs and the scheme's capabilities (Marcolla, 2022). For example, most HE schemes are designed to operate in applications that use a single, common key. Hence, implementation barriers may arise if multiple keys are used to encrypt the input dataset. This means, for example, that if an averaging function is to be applied to a dataset that aggregates data from multiple users, and each user record is protected by a unique user key, then HE mechanisms cannot be used without introducing additional logic (such as *Proxy ReEncryption*).

Proxy ReEncryption (PRE) is a computing technique that uses an intermediary or proxy entity to convert the ciphertext of one user (delegator) into a ciphertext of another user (delegatee). The delegator's key is not exposed to the delegatee, but the delegatee can still successfully decrypt the converted ciphertext. By not exposing the delegator user's keys or plaintext, the proxy mechanism ensures privacy is maintained. However, there exists a risk that the proxy replaces the data or uses a deliberately weak key without the delegator knowing. Hence, there must exist an implicit level of trust towards the intermediary. Such risks may not always be manageable. For example, although the original PRE method incorporated mechanisms to mitigate against these types of risks (Craig, 2009), later analysis revealed that it was vulnerable to collusion attacks (Marcolla, 2022). This means the delegator's private key can be left exposed if the delegatee and proxy collude. Subsequent research introduced

Hierarchical Proxy Re-Encryption (HPRE) schemes that are resilient to weak collusion[1] attacks but are still vulnerable to strong collusion attacks. Another known issue of HPRE schemes is that they are only secure against chosen plaintext attacks (CPA), which lowers the number of scenarios they're appropriate for.

One security issue with all known FHE schemes is that the bootstrapping process requires access to the data generator's public key, which may introduce cybersecurity risks for entities that outsource the encryption process (e.g., to cloud-based services). Although sharing a public key in public key cryptography does not reveal the original data, its use is still considered a subject of debate (Leluc, Chedemail, Kouande, Nguyen, & Andriamandratomanana, 2022). Another vulnerability of the unbounded FHE schemes is that they are vulnerable to indistinguishability under chosen ciphertext attack 1 (IND-CCA1)[2].

2.3 Computational Overhead of Homomorphic Encryption Schemes

Due to the significant amount of computational overhead incurred in the use of HE, several literature surveys have been carried out in order to better characterize the processing delays, key lengths, and message sizes required to implement a feasible HE solution. One such survey was presented in (Doan, 2023). Among the PHE schemes reviewed, the list included *Rivest-Shamir-Adleman* (RSA), *Paillier*, and *El-Gamal* schemes. As summarized during the introduction, RSA enables unlimited multiplicative operations to be performed, whose security relies on the complexity of solving the prime number factoring problem. *Paillier* allows an unlimited number of additive computations and is based on the composite residuosity problem. The *El-Gamal* scheme is also multiplicative, and its security relies on the difficulty of solving discrete logarithms.

The paper also dives into more modern (and complex) algorithms that are based on the R-LWE problem assumption. This includes the *Cheon-Kim-Kim-Song* (CKKS) scheme, which is a SWHE scheme that can approximate results of additive and multiplicative operations on complex numbers; as well as other similar schemes, such as the *Brakerski-Fan-Vercauteren* (BFV), *Brakerski-Gentry-Vaikuntanathan* (BGV), *Fast Fully Homomorphic Encryption* (TFHE), and *Fastest Homomorphic Encryption in the Wes*t (FHEW) schemes.

All HE algorithms were executed using applicable HE libraries on a computer equipped with an Intel(R) Core(TM) i7-10700 CPU running at 2.90GHz under Ubuntu 20.04 (Doan, 2023). The computational overhead time (in microseconds) was calculated from the average time taken by the operation over the span of 1,000 iterations. The encryption operation time encompasses the duration taken to generate random values for message inputs, along with the time required for encoding and decoding the batches (Doan, 2023). All algorithm parameters were selected to provide the equivalent of a 128-bit encryption security level and bootstrapping was not applied to FHE algorithms.

[1] Collusion attacks occur when multiple parties pool their information to gain unauthorized access to the encrypted data. The term weak is used to describe when a non-majority number of agents collude.

[2] In IND-CCA1, the adversary obtains a public key from the verifier. The adversary sends two different plaintexts to the verifier, who randomly encrypts one of them and sends back the result (referred to as the challenge). The goal of the adversary is to determine which of the plaintexts were encrypted. The adversary can send an unlimited number of ciphertexts to a decryption oracle to decrypt before the challenge is sent, after which, queries are disallowed. A system that is IND-CCA1 secure ensures that an adversary cannot accomplish their goal with a probability significantly higher than 50%.

The time overheads and space requirements (encrypted message size) were obtained for various HE algorithms using the *Homomorphic-Encryption Library* (HeLib), PALISADE, *Simple Encrypted Arithmetic Library* (SEAL), and the *Homomorphic Encryption for Arithmetic of Approximate Numbers* (HEAAN) open-source libraries. HeLib uses BGV and CKKS schemes with ciphertext packing techniques and Gentry-Halevi-Smart optimizations. PALISADE supports the BGV, BFV, CKKS, FHEW, and TFHE schemes, with multi-party extensions for some HE schemes. SEAL supports the BFV, BGV, and CKKS schemes, and it contains optimizations intended to lower its initial learning curve. HEEAN supports only the CKKS scheme and provides addition and multiplication operations on fixed-point values and approximate operations on rational numbers. The PHE schemes tested were implemented by the researchers themselves (Doan, 2023). Figure 2 and Figure 3 illustrate the results of the various operations performed by the PHE schemes tested by (Doan, 2023). The data used to obtain the graphs can be viewed in Table 3-Table 5 (also computed by (Doan, 2023), but reproduced in this report for convenience). In Table 3 and Table 4, P is the plaintext modulus, N is one factor in the encryption keys, and $\log_2(N)$ is the number of bits in N.

Table 3: Multiplicative PHE Scheme Comparison From (Doan, 2023)

HE Parameters						
Scheme	P	$Log2(N)$	KeyGen	Enc	Dec	Mult
RSA	1032193	109	1484.327	1.062	5.399	2.237
El-Gamal	1032193	109	31063.45	4.486	4.336	3.151
RSA	1032193	218	1931.446	1.669	7.803	0.396
El-Gamal	1032193	218	135618.53	16.748	15.476	9.74
RSA	786433	438	3490.179	2.496	37.265	0.853
El-Gamal	786433	438	773368.775	66.794	32.56	18.388
RSA	786433	881	8366.837	6.865	205.38825	2.178
El-Gamal	786433	881	5354554.333	403.448	203.7188	8.651
RSA	$\mathbb{Z}_n$	3072	180255.34	61.599	6327.537	2.934
El-Gamal	$\mathbb{Z}_n$	3072	>15 minutes			
RSA	$\mathbb{Z}_n$	4096	433348.8	88.372	14327.857	9.792
El-Gamal	$\mathbb{Z}_n$	4096	>15 minutes			

Table 4: Paillier PHE Scheme Time Complexities From (Doan, 2023)

HE Parameters (for Paillier)					
P	$Log2(N)$	KeyGen	Enc	Dec	Add
1032193	109	1072.014	265.509	6.738	4.255
1032193	218	1537.688	279.664	22.872	4.053
786433	438	3081.14	367.893	141.012	6.08
786433	881	7903.175	1013.957	950.735	10.868
$\mathbb{Z}_n$	3072	183774.01	20237.659	26364.306	232.136
$\mathbb{Z}_n$	4096	4297885.6	42868.889	55843.731	212.978

In Figure 2, t is the time in microseconds. In Figure 2 and Figure 3, N is one factor in the encryption keys, and $log2(N)$ is the number of bits in N.

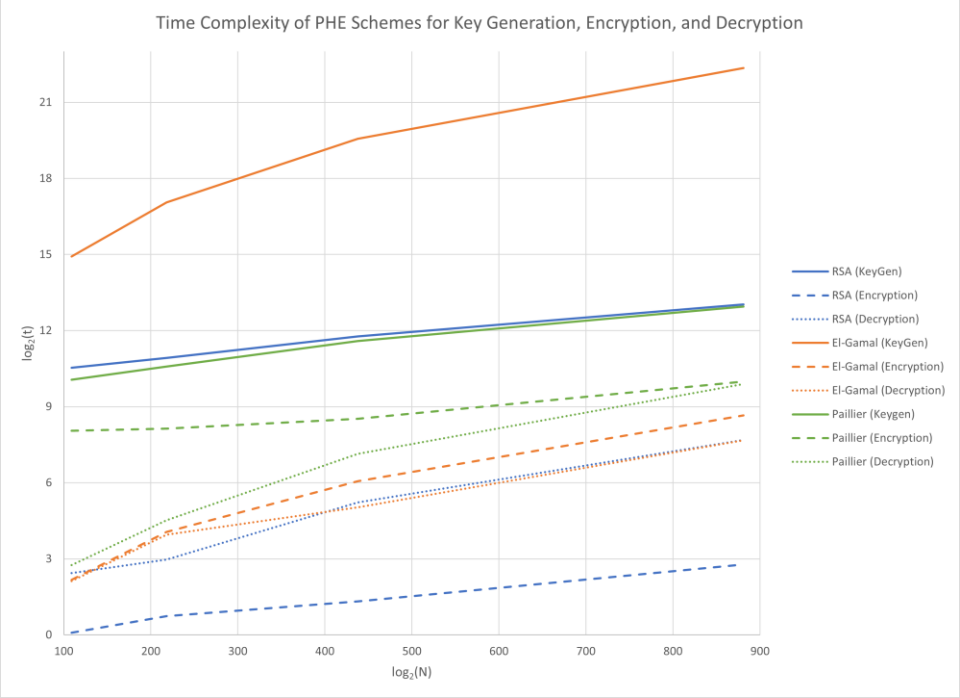


Figure 2: Time Complexity of PHE Schemes for Key Generation, Encryption, and Decryption Computations

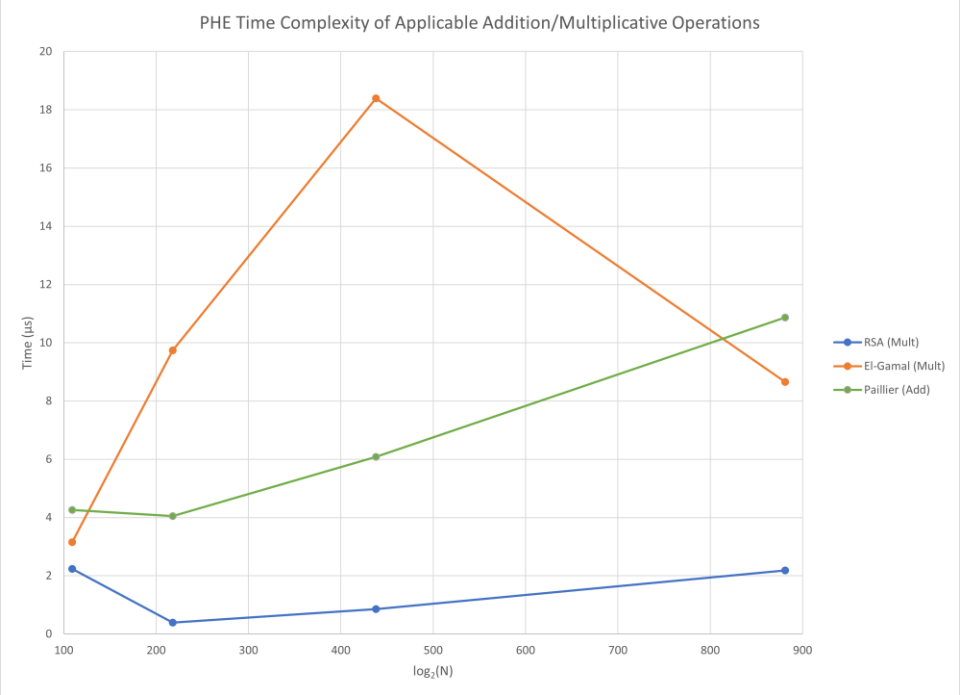


Figure 3: Time Complexity of PHE Schemes for Addition/Multiplication Operations

As shown in Figure 2, Paillier and RSA have similar key generation times when $N < 900$, with Paillier being slightly faster than RSA. However, when $N > 900$ and a 128-bit level of

security is required (as seen by viewing Table 3 and Table 4), Paillier becomes much slower at key generation than RSA, while El-Gamal takes longer than 15 minutes. For example, when $log2(N)$ is 3,072, Pailler takes 3,518.67 μs longer than RSA, and when $log2(N)$ is 4,096, Pailler takes ≈3.864 seconds,536.8 μs longer than RSA. RSA performs encryption the fastest among the three PHE schemes presented. RSA and *El-Gamal* have similar decryption speeds when a security level of 128 bits is not guaranteed. When 128-bit level of security is guaranteed, the time for key generation, encryption, and decryption all increase in *Paillier*. For example, when N is 4096 bits, generating a single key takes nearly 4.3 seconds. When the value of $log2(N)$ is 881, *El-Gamal* takes longer than 5 seconds to generate key pairs. As displayed in Figure 3, RSA outperforms *El-Gamal* for speed of multiplicative operations.

In Table 5 – Table 7, N is the dimension of the ciphertext, and Q is the maximum ciphertext modulus.

Table 5: BFV Scheme Comparison From (Doan, 2023)

Library	HE Parameters						
	N	$Log2(Q)$	KeyGen	Enc	Dec	Add	Mult
SEAL	4096	109	1028.119	1263.528	276.045	1.298	3274.257
PALISADE	4096	120	1137.556	1160.459	283.99	0.237	4296.438
SEAL	8192	218	3003.509	3269.548	1179.682	144.531	11663.16
PALISADE	8192	180	3170.82	2881.717	921.646	187.703	13585.75
SEAL	16384	438	10260.45	11378.441	5434.016	415.662	54918.967
PALISADE	16384	420	13507.743	11288.5535	3298.9775	1086.105	76565.506
SEAL	32768	881	40251.496	41297.274	17442.857	1536.587	246427.201
PALISADE	32768	840	55941.007	45587.262	17171.713	7046.362	427795.343

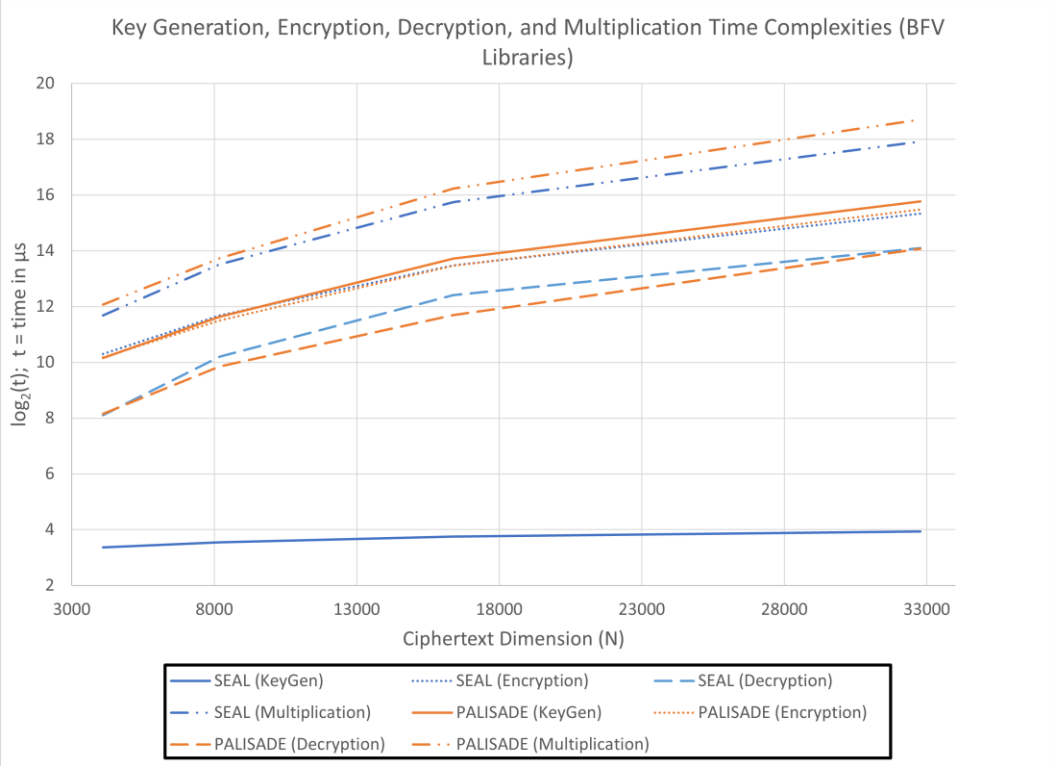


Figure 4: Time complexities associated with *Key Generation*, *Encryption*, *Decryption*, and function computation (multiplication) via the BFV Libraries

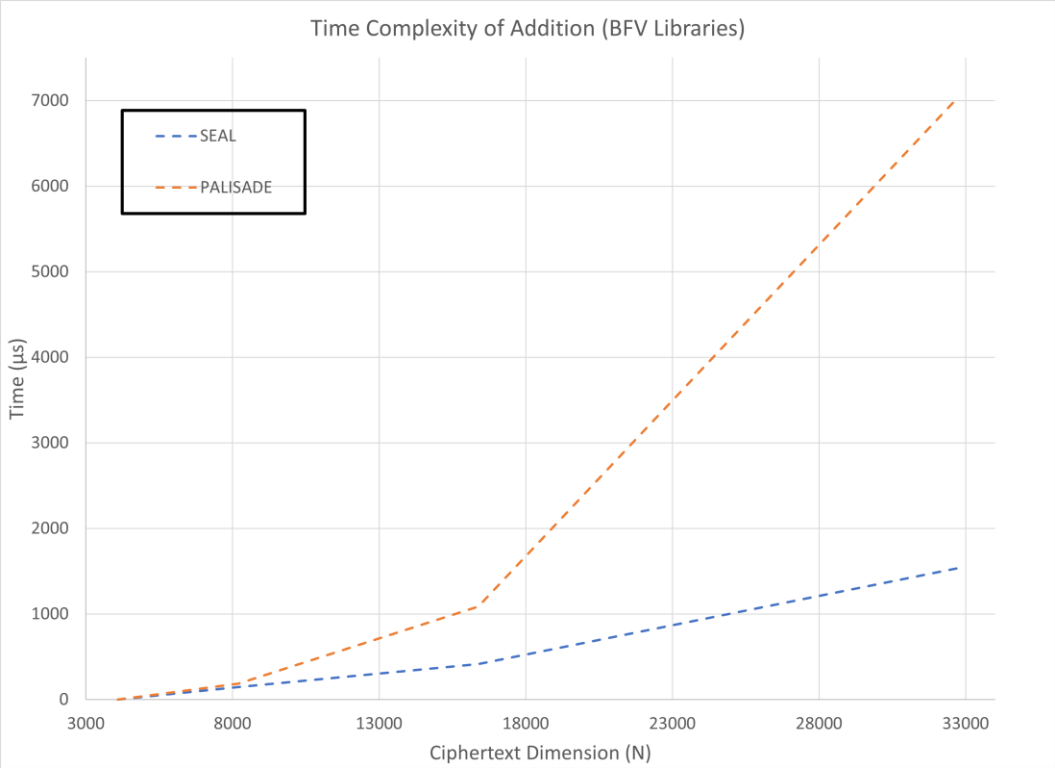


Figure 5: Time Complexity of Addition (BFV Libraries)

As the dimension of the ciphertext N increases, the time to execute the operation increases as well. For both libraries, each time the current ciphertext dimension N is doubled, the multiplicative operation takes approximately four times longer. For example, when $N = 4096$, SEAL takes 3274.257 μs to perform multiplication, and when $N = 8192$, SEAL takes 11663.16 μs. The average execution time of the multiplicative operation in SEAL is less than PALISADE because SEAL performs the re-linearization step as a separate computation, while PALISADE

always counts the re-linearization step as part of the multiplication function. Overall, SEAL does better than PALISADE in most cases with regards to time complexities (except for decryption).

Table 6 BGV Scheme Comparison From (Doan, 2023)

	HE Parameters						
Library	N	$Log_2(Q)$	KeyGen	Enc	Dec	Add	Mult
SEAL	4096	109	2424.838	1091.586	259.842	42.541	1509.681
PALISADE	4096	96	3023.297	1145.76	368.375	42.116	570.952
HElib	4096	100	168300.764	2257.432	138092.51	32.064	2347.865
SEAL	8192	218	11426.94	3137.433	992.5	79.952	6673.09
PALISADE	8192	144	10981.757	3043.417	1007.424	57.322	2396.688
HElib	8192	100	470367.195	4533.877	549616.633	480.44	4492.487
SEAL	16384	438	70869.416	11179.579	3791.998	292.17	35650.547
PALISADE	16384	240	51708.9	8902.513	3547.961	289.751	13642.014
HElib	16384	100	1348552.91	9917.878	2265994	289.706	10778.79
SEAL	32768	881	433638.89	41716.827	18156.642	866.2635	215414.681
PALISADE	32768	480	376273.9767	34662.558	20727.674	3313.547	116248.311
HElib	32768	100	1967110.87	14080.4445	2340201.2	209.039	17477.661

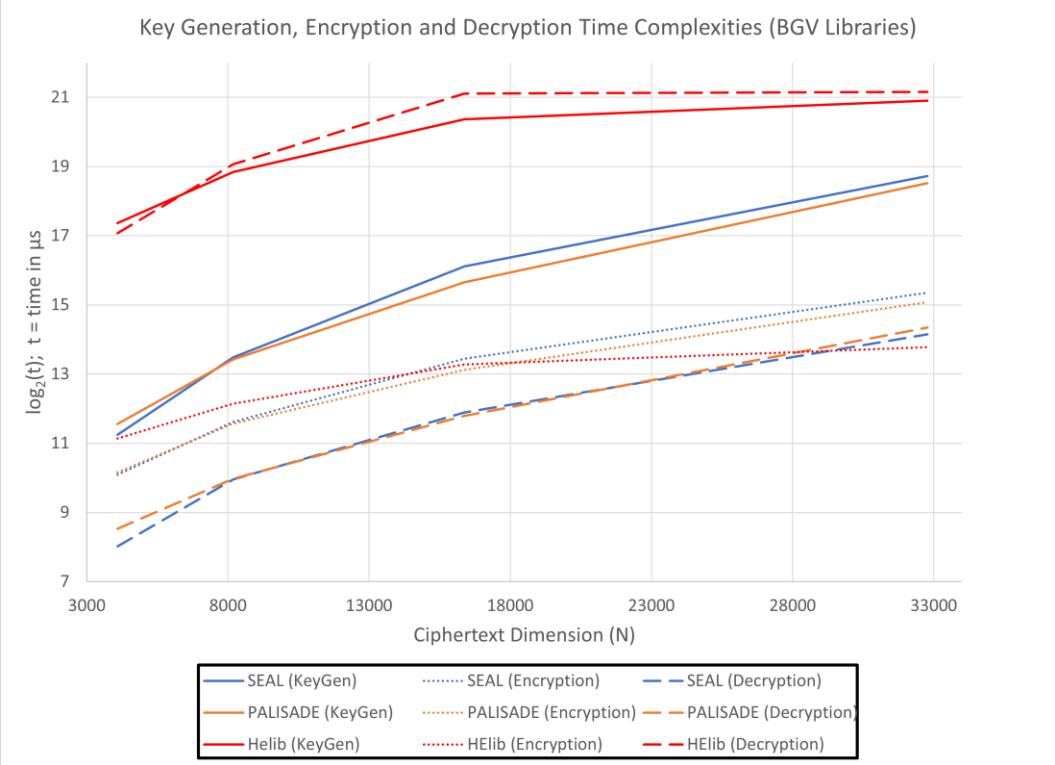


Figure 6: Key Generation, Encryption and Decryption Time Complexities (BGV Libraries)

Figure 7: Time Complexities of Addition and Multiplication (BGV Libraries)

SEAL and PALISADE perform much better than HElib for key generation and decryption. HElib offers better execution times over SEAL and PALISADE for multiplication. HElib also overtakes them in encryption and addition execution times as the dimension of the ciphertext increases.

Table 7: CKKS Scheme Comparison From (Doan, 2023)

	HE Parameters						
Library	N	$Log_2(Q)$	KeyGen	Enc	Dec	Add	Mult
SEAL	8192	200	2507.607	3910.8775	109.5735	271.207	452.792
PALISADE	8192	102	2305.699	2652.975	21650.503	81.117	3129.505
HElib	8192	(119, 157.866)	11008.069	2659.019	22065.082	272.865	19712.186
HEAAN	8192	119	2282102.44	634268.04	41491.42	39877.65	614878.85
SEAL	16384	432	11959.254	18847.575	721.076	1777.956	2077.872
PALISADE	16384	141	6542.385	7093.977	51639.085	194.05	9584.286
HElib	16384	(358, 129.741)	91768.896	8252.838	107935.827	1502.701	104850.697
HEAAN	16384	358	2294477.86	624440.22	93658.41	17826.4	994892.6
SEAL	32768	881	39749.74	66061.559	2589.904	2221.2535	4741.014
PALISADE	32768	342	31630.72	29936.449	248985.192	3291.783	66603.916
HElib	32768	(558, 128.851)	164575.383	23730.201	364743.317	11576.171	215878.991
HEAAN	32768	558	2251482.12	657943.99	114587.91	45690.59	1332368.41

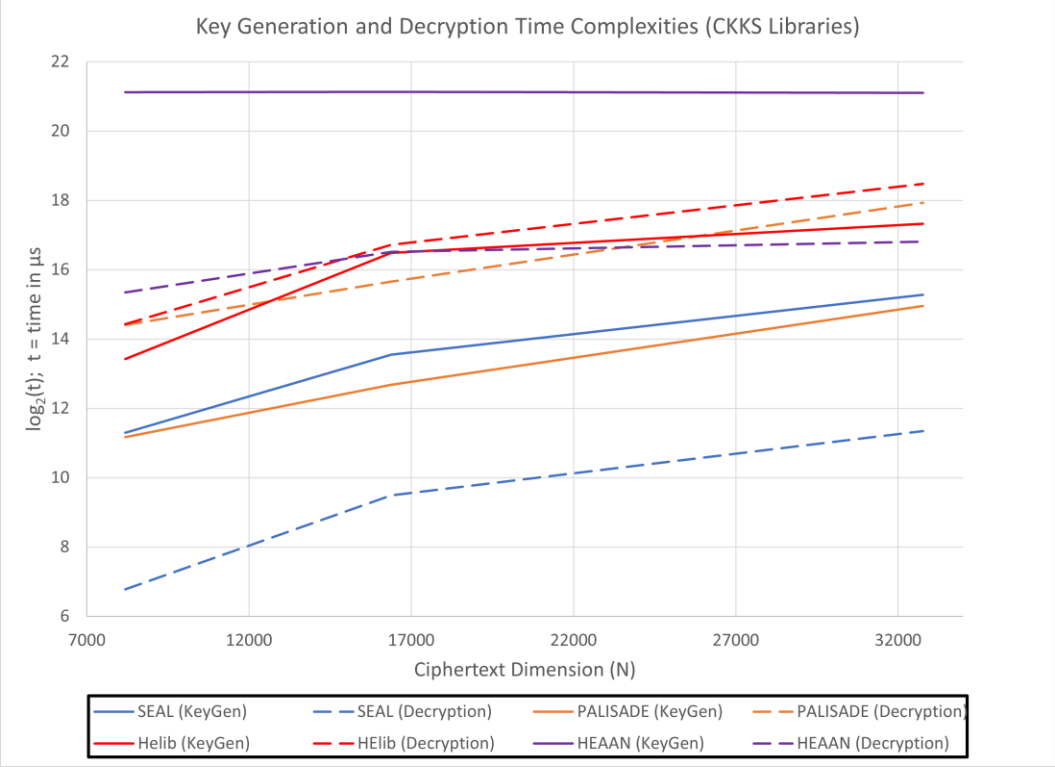


Figure 8: Key Generation and Decryption Time Complexities (CKKS Libraries)

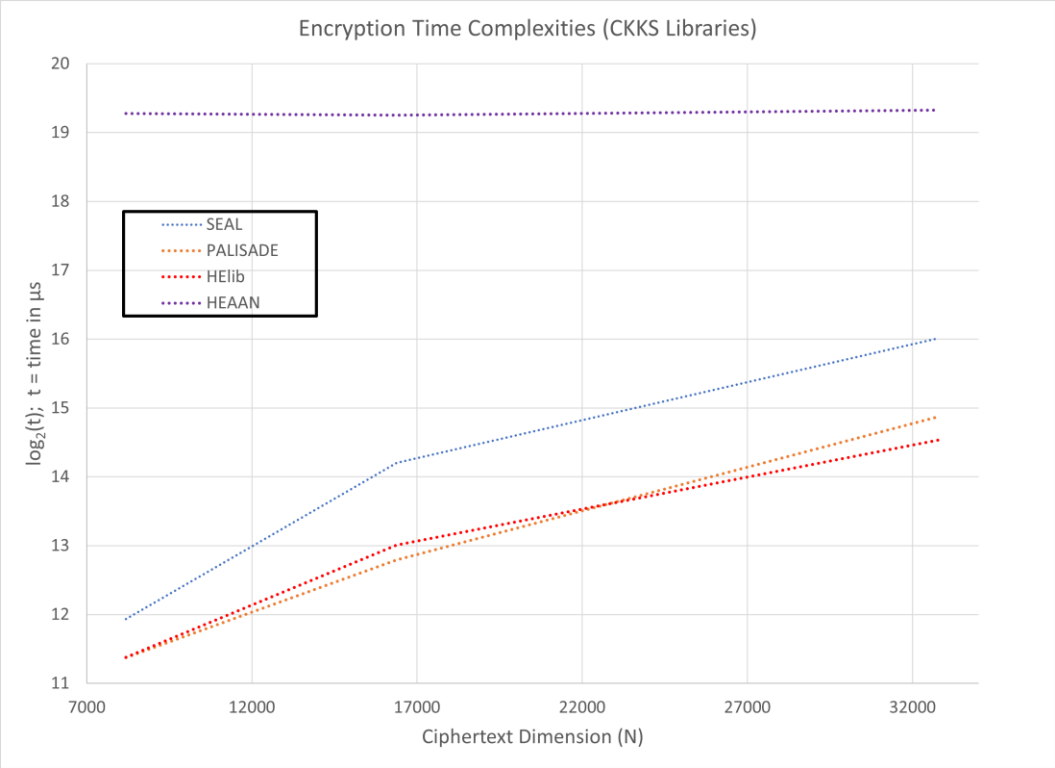


Figure 9: Encryption Time Complexities (CKKS Libraries)

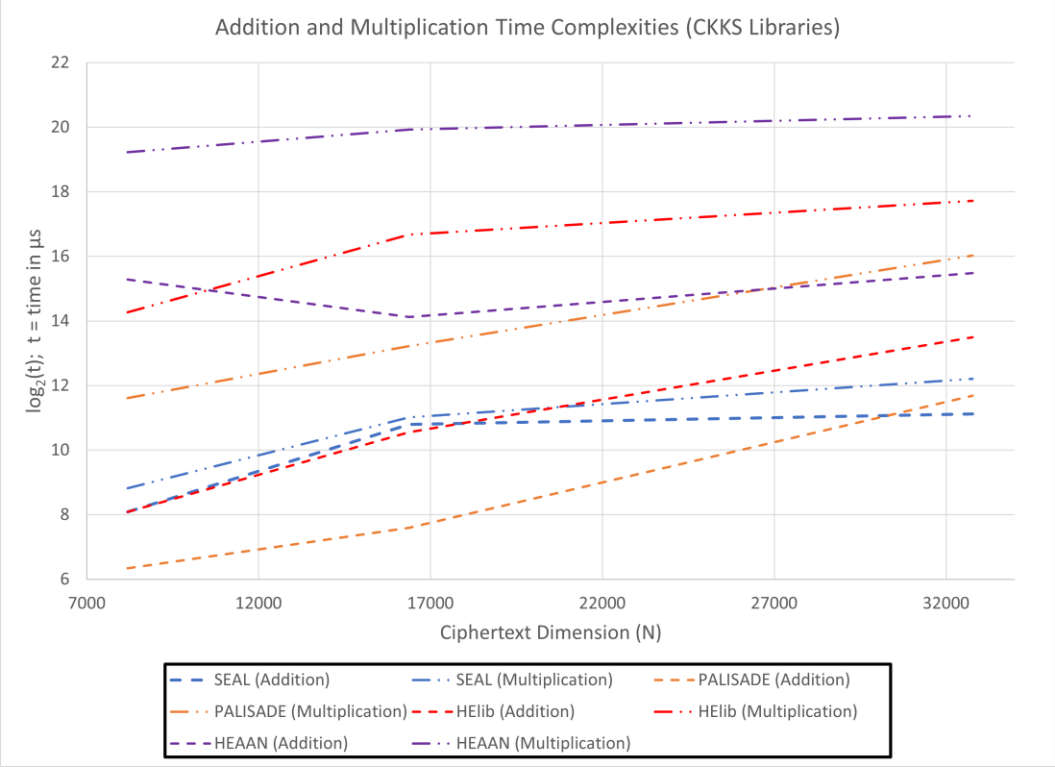


Figure 10: Addition and Multiplication Time Complexities (CKKS Libraries)

HEAAN does far worse than the other schemes on all operations except for decryption when the dimension of the ciphertext being decrypted is larger than $\approx 2x10^4$. SEAL and PALISADE perform similarly on key generation, addition, and multiplication operations. However, SEAL offers the best decryption and multiplication times, as well as addition when the dimension of the ciphertext is greater than $\approx 2.6x10^4$. Overall, SEAL performs better than the other libraries when using the CKKS algorithm.

The computation times for key generation, encryption, decryption, addition, and multiplication operations between BFV, BGV, and CKKS schemes are shown in Figure 11 and Figure 12. The data for Figure 11 and Figure 12 were obtained from the Microsoft SEAL library entries in Table 5 – Table 7. The Microsoft SEAL library was chosen to take the measurements from because it performed better overall than the other libraries for two of the three schemes. For the BFV scheme, it only performs worse than Palisade in decryption times. For the CKKS scheme, Microsoft SEAL performs similarly to or better than Palisade, which are both better than the other libraries for most operations except decryption, where Microsoft SEAL performs better than all other libraries tested. There is no single library that performs better than the rest for the BGV scheme, thus, the Microsoft SEAL data was used for consistency. While BFV and BGV can obtain a security level of 128-bits with a ciphertext dimension of 4096, CKKS requires a ciphertext dimension of 8192. Thus, the ciphertext dimension of 4096 is omitted from Figure 11 and Figure 12.

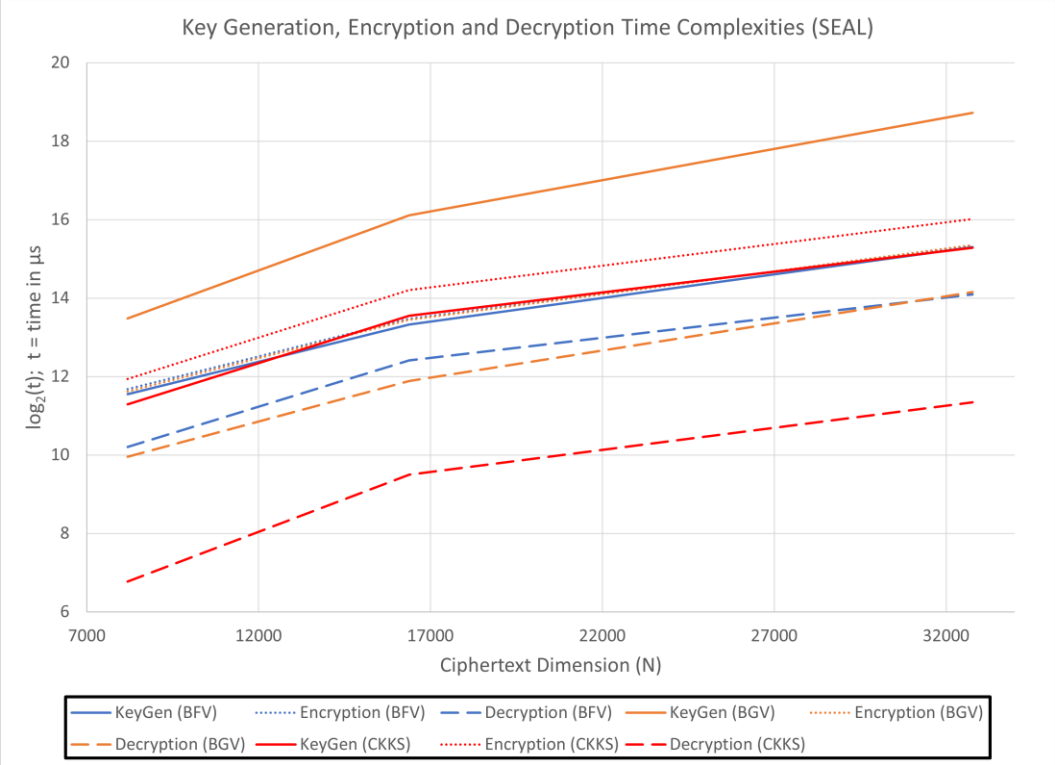


Figure 11: Key Generation, Encryption and Decryption Time Complexities of BFV, BGV, and CKKS Schemes Using SEAL Library

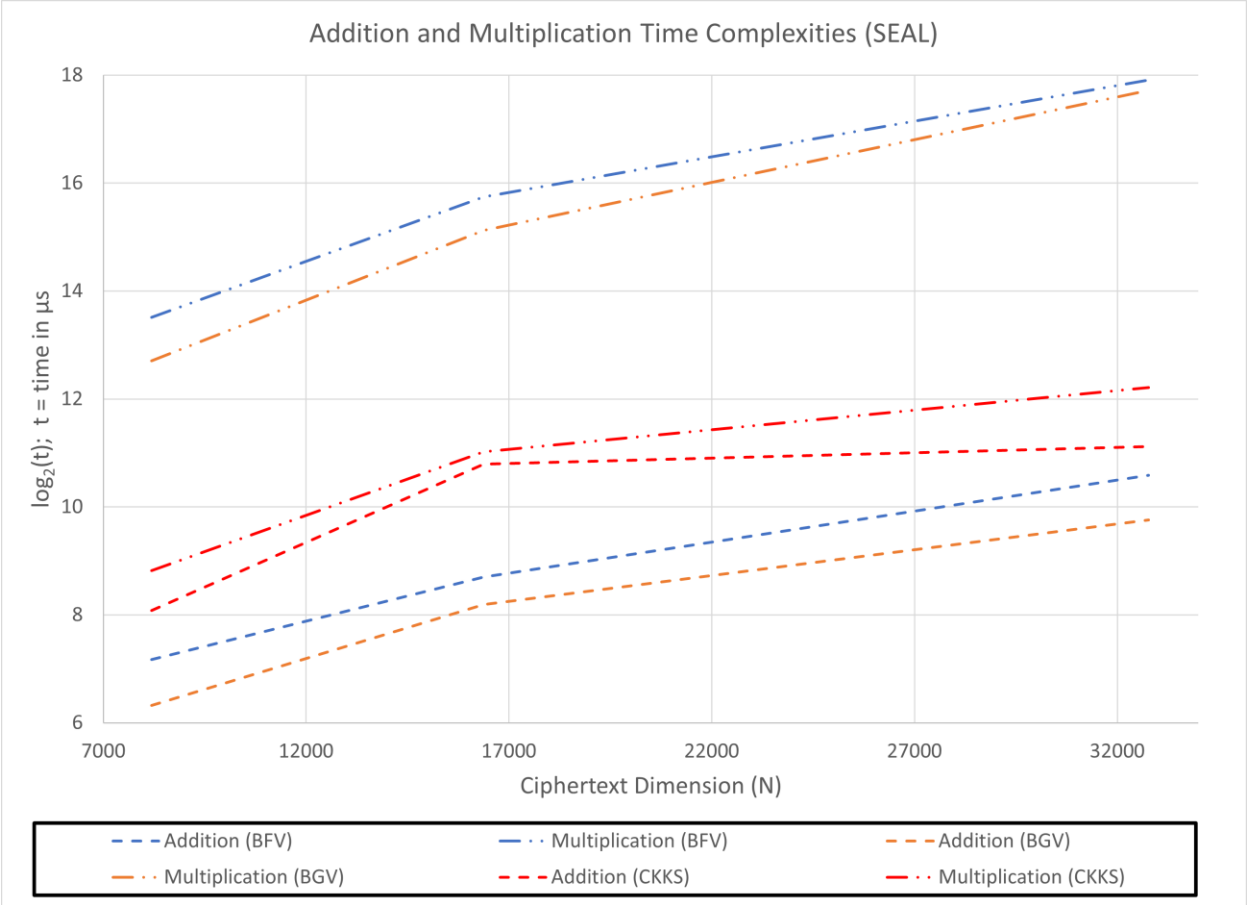


Figure 12: Time complexities associated with the Addition and Multiplication functions under the BFV, BGV, and CKKS Schemes (SEAL Library)

BFV and CKKS have similar key generation times that are better than BGV. BFV and BGV have similar encryption execution times that perform better than CKKS, however CKKS is much better at decryption times. BGV offers the best performance involving addition, while CKKS is superior to the other schemes involving multiplication. No scheme has a clear overall advantage over the others. CKKS does the best overall with regards to execution time if the total number of operations it executes quicker than or performs similarly to (time within ± 2,000 µs) the other schemes is considered.

3.0 Zero-Knowledge Proofs

A zero-knowledge proof (ZKP) is a verification protocol that allows one entity to prove to another entity that they have knowledge of secret data without revealing anything besides the fact that they have that knowledge (Sun, 2021). The entity providing the proof is known as the prover, and the entity they are trying to convince is known as the verifier. The framework for a ZKP scheme consists of three phases:

1. Witness Phase – The prover sends a calculated proof containing its statement to the verifier.

2. Challenge Phase – The verifier sends questions to the prover.

3. Response Phase – The prover sends back responses to the questions to the verifier, who uses them to accept or reject the proof.

The protocol has the properties of *completeness*, *soundness*, and *zero-knowledge*. The *completeness* property states that if the prover's statement is true and they can prove it, then the verifier will always accept it. The *soundness* property states that if the prover's statement is false, then there is only a small probability that they will be able to convince the verifier that it is true. The *zero-knowledge* property provides the guarantee that the prover does not expose any information to the verifier other than the fact that their statement is true (See Annex B.3 for a numerical implementation of ZKPs).

A ZKP scheme can be interactive or non-interactive (Deng, 2019). In an interactive ZKP, the prover and verifier need to communicate a minimum of three times to complete the necessary challenge and response phases. Since the two entities must communicate with each other, there is no support for offline operation, which causes the algorithm to be inefficient. An interactive ZKP can be converted into a *Non-interactive ZKP* (NZKP) through a heuristic technique (e.g., Fiat-Shamir). A NZKP removes the challenge phase from the framework (Sun, 2021). For example, in the *Zero-Knowledge Succinct Non-Interactive Argument of Knowledge* (zkSNARK) scheme, a trusted third party generates proving and verification keys that are sent to the prover and verifier respectively. These keys are used to construct the proof and verify it without the need for the verifier to ask the prover further questions. This reduces execution time and enables offline verification via asynchronous communication between the prover and verifier. More advanced methods, such as NZKP's (e.g., Bulletproofs) have further removed the need for a trusted third party to setup the keys.

A common use for ZKPs is providing verifiable proof for the exchange of a tangible good (Sun, 2021). A ZKP can be utilized to verify that the prover has enough resources to complete a request through a process that can be summarized as follows:

1. A proof is created with the prover's claim that they have enough resources and is sent to the verifier.

2. The verifier applies predefined computations on the proof to obtain a decidable outcome and evaluates whether the statement is accepted or rejected.

No information about the prover's specific resources is revealed by the interaction. Either the prover doesn't have the necessary resources and the value becomes negative, or they do have the resources and the value remains positive.

3.1 Benefits and Applications of Zero-Knowledge Proofs

Due to their self-attestation capabilities, ZKPs provide a number of advantages when compared against traditional authentication schemes that depend on third-party attestation mechanisms, such as those typically found in public key infrastructure (PKI) solutions (Chen, 2023). For example, the zero-knowledge property allows entities to maintain anonymity by removing the need for an authoritative registration process, while still enabling individuals to prove their identity (or other intrinsic property) to their peers. This capability opens the door to a wide variety of potential applications that range from being able to verify past behaviors to proving new functions or capabilities without publicly disclosing how these will be accomplished. Although some of these features could be implemented using traditional methods, a ZKP approach naturally enables the creation of decentralized systems, which tend to simplify the challenges associated with deploying and maintaining large-scale systems. This unique capability enables ZKPs to be well positioned to address the operational needs that are typically associated with managing and operating large, autonomous sensor networks. ZKPs also help to reduce the attack surface associated with traditional authentication schemes because they do not store any authentication data that could be stolen. These characteristics enable ZKPs to avoid most of the common attacks associated with authentication-based systems (e.g., identity impersonation) because instead of binding users to credentials, it relies on internal properties to assemble a credible proof that reflects the original subject. Another advantage of ZKPs is that they have a relatively low overhead when compared to PKI-based systems, a feature that may be relevant in resource constraint systems. Furthermore, there are specialized techniques available to lower the complexities associated with the computation and communication stages, such as reducing the proof lengths, which may be useful in Internet of Things (IoT) environments because it removes the need for managing a large (and often complex) identity and authentication management (IAM) system, a challenge that is further compounded when millions of IoT enabled devices must operate over the same network.

Due to their capabilities, ZKPs have been proposed as a viable tool to address the challenges associated with implementing a wide array of applications that range from anonymous voting applications to securing the exchange of digital assets, securing biometric authentication, and creating secure auction platforms (Sun, 2021). Although each case is unique, the typical use case seeks to decouple the identity and private attributes from a reportable attribute, thereby ensuring privacy and security while providing means to validate and optionally quantify the attribute under scrutiny. For example, in a voting application, the identity and the associated vote must be kept protected while still enabling the vote to be counted towards the net tally.

Another frequently cited example that benefits from adopting ZKP methods is on the implementation of digital asset exchange platforms that aim to maintain privacy. This may include preventing users from having to disclose their true identity, to providing mechanisms that prevent others from monitoring past activities or determining the assets being held (Sun, 2021). A correct implementation of ZKP could hence provide the necessary proofs to validate that an asset exchange has taken place without further data leakage. Some relevant examples of such applications include Zerocash, Hawk, and Bolt which operate over blockchain networks to ensure exchange records are permanently recorded while also providing support for automated transactions. In particular, Zerocash aims to protect the private information of the transaction amount, source, and destination by supporting anonymous transactions. In the Hawk scheme, the final outcome of the smart contract can be verified with NZKP, but the actual transaction history is kept hidden. Bolt offers a bidirectional channel for secure payment that utilizes ZKP and blind signatures to avoid exposing the linkage of the users' payments. To

further illustrate the wide diversity of applications on which ZKP can be applied, Table 88 provides a high-level summary of various research efforts found in literature.

Table 8: Applications of ZKP Across Various Domains

Citation	Keywords	Overview
(Gaba, 2022)	Internet of Healthcare Applications (IoHA), Internet of Things (IoT), ZKP-based authentication and key agreement protocol	The researchers developed a ZKP-based *Authenticated Key Agreement* (AKA) protocol for IoHA. In the protocol, the healthcare professional (user) and the IoT sensor node are registered using a ZKP algorithm. During the process, temporary identities are given to the user and sensor node to maintain anonymous communication. When the user wants to exchange data with the sensor node, the legitimacy of the user, user device, sensor node, and gateway are first validated. The benefits of the proposed scheme are that it provides untraceable and anonymous communications in the public channel and withstands attacks such as man-in-the-middle, replay, and impersonation.
(Singh R. A.-W., 2023)	Privacy-Preserving Healthcare Financial System, Blockchain	A decentralized healthcare finance system based on blockchain (DLT) and ZKP was proposed by the researchers to preserve the privacy of the system user's data. The data exchanged includes transaction amounts and the identities of the parties involved. Several ZKPs used in their system to validate values are within a certain range, or have a certain balance, while also providing proof of consistency to ensure the correctness of transactions and eliminate the double spending problem. The proposed scheme enables basic statistical queries (e.g., sums, averages, variance, and skewness) without leaking additional information. The validation of the transactions is stated to be completed in milliseconds.
(Liu J. K.-K., 2020)	Privacy-Preserving COVID-19 Contact Tracing, Healthcare`	The researchers proposed a ZKP-based protocol to perform COVID-19 contact tracing while protecting locational and personal contact information. The protocol consisted of four phases: registration, meeting, medical treatment, and tracing. In the first phase, users upload their public keys. In the next phase, the user's smartphone periodically broadcasts a beacon. Once enough beacons are received, a pairing occurs that mutually validates and generates a credential that proves that a close contact occurred. In the medical treatment phase, the proof is presented to a doctor who is unable to learn their identities, or other private attributes. The doctor signs the ZKP using their secret key and posts the signature and proof to a public bulletin board. Finally, in the tracing phase, each user periodically checks the bulletin board for exposure tracing.
(Liu S. G., 2023)	Power System, Anonymous Authentication	The researchers developed an anonymous authentication protocol to preserve the real identity of users in a power system. The protocol combined elliptic curve cryptography (ECC) and the Fujisaki-Okamoto commitment protocol. The framework consisted of registration, mutual authentication, and revocation stages. ECC was used in the registration and authentication stages to handle the large amount of data in the power system. A certificate authority is used in the revocation phase. The researchers provide proofs of user anonymity, mutual authentication, validity of anonymity, and immunity to replay attacks.
(Pop, 2020)	Blockchain (DLT), Prosumer Energy Data	A decentralized implementation of a demand response program based on blockchain and ZKPs was presented by the researchers. ZKPs were utilized to obscure the prosumer's energy data and

	Privacy, Demand Response,	requested flexibility profiles. The blockchain stored the deviation quantities and performed ZKP validation to verify the deviation was correctly computed. The researchers determined that their scheme was secure against malicious entities that are a part of the blockchain network because the actual energy values are not stored on the blockchain. The scheme may also improve the demand response program's audit and financial settlement, which could lead to implementation of micro-grid level consensus algorithms.
(Yang R. H., 2024)	Virtual Power Plant, Attribute-Based Encryption, Attribute Hiding	The researchers proposed an attribute-hiding ZKP (AH-ZKP) to conceal the identities and attributes of users during the authorization and authentication processes found in a virtual power plant (VPP) system. Potentially sensitive information that can be extracted from VPP systems includes customer information, power consumption values, electrical equipment specifications (e.g., type, brand, model), and energy usage patterns. The researchers performed a security analysis on the scheme and determined that their scheme is robust against man-in-the-middle and replay attacks and offers attribute indistinguishability and tamper resistance. Comparative experimental analysis found that the attribute-based encryption and ZKP steps added a 1 second computation burden, which outperforms existing methods in time and space efficiency.
(Gabay, 2019)	Electric Vehicles (EV), EV Charging Process Privacy Preservation, Blockchain	A privacy-preserving, decentralized EV scheduling tool based on ZKP was proposed by the researchers. The proposed approach addresses the challenges associated with centralized EV Service Providers (EVSP), which could enable access to the driver's personal information, travel patterns and habits. In their scheme, the EVSP generates a secret function and a proving key that is given to the EVs. The EV uses the proving key and result from solving the secret function to create a proof that is presented to an Ethereum (blockchain) network for authentication. A smart contract then generates a service-token that can be used for anonymous scheduling. According to the researchers, time and cost overheads were acceptable for real-life applications.
(Ho, 2021)	Smart Parking, Zero-Knowledge Set Membership Proof, Bluetooth Low Energy (BLE) Beacon	The researchers retrofitted ZKP to an on-street parking system, enabling users to maintain privacy by anonymously authenticating with the server. The ZKP was combined with a commitment scheme and Merkle tree to accomplish this goal. Users first register their vehicle data and national ID, along with the hash of their public key with the authority server, which is stored in an identity commitment Merkle tree. To authenticate, users query the server for the values required to construct the proof. The proof is sent to a nearby roadside unit, who interacts with the server to verify the proof and approves the parking request. The researchers determined their new scheme improved the robustness of their previous system against identity forgery, replay attacks, and masquerade attacks.

3.2 Challenges Faced by Zero-Knowledge Proofs

As with other engineering tools, ZKPs have become highly specialized, resulting in a wide variety of implementations that seek to solve specific challenges. For example, the *Zero-Knowledge Range Proof* (ZKRP) scheme provides a verifiable proof that a value (integer or binary) lies within a specific range (Morais, 2019). Under the ZKRP scheme, integer proofs can be achieved by leveraging two main methods that are based on the square decomposition problem and digital signatures. The first approach breaks down the secret value into a sum of

squares while with the latter approach, each element within the interval is signed, enabling the prover to provide a proof that demonstrates knowledge of the signature (Boudot, 2000). For binary values, multi-base decomposition and two-tiered homomorphic commitments are used. In multi-base decomposition, the secret value is converted into its bit representation and Boolean arithmetic is used to verify that it belongs to a specific interval. ZKRPs have applications in age validation, mortgage risk assessment, electronic voting, electronic auctions, and asset procurement (Morais, 2019). For example, a ZKRP can be used to verify that a user's age is higher than some value (18, 21, etc.) without revealing their actual age, or to verify that an individual's salary meets the mortgage requirements.

As many other technical constructs, ZKP methods can be designed or at least optimized to satisfy specific application needs, such as making them more efficient for real-time communications. Despite the benefits of specialization, improperly applied optimizations can result in poor security capabilities (Tang, 2024). For example, several ZKP mechanisms rely on the Fiat-Shamir Transformation to achieve NZKP functionality. However, there is limited guidance on the secure implementation of the Fiat-Shamir Transformation protocols, which can lead to flaws and vulnerabilities in the system. As an example of such a vulnerability, researchers have identified "Frozen Heart" in a number of implementations (e.g., PLONK, Bulletproofs) (Tang, 2024). Said vulnerability enables a malicious actor to forge proofs that pass all verifications regardless of the range being queried, essentially rendering ZKP useless (by violating the *soundness* requirement).

On the other hand, some schemes significantly change the way that ZKP operates and have re-introduced requirements that significantly differ from the ZKP foundational concepts. For example *Ligero* has re-introduced third party attestation in order to increase efficiency (Sun, 2021). Another challenge that is often encountered is on how to merge two distinct schemes and combine them to achieve an improved ZKP scheme. For example, the running time of the prover algorithm is linear in the Libra scheme (Sun, 2021) but requires a trusted third-party setup. In contrast, the Hyrax scheme does not require a trusted third party, but the prover algorithm runs in loglinear[1] time complexity (Sun, 2021).

In the domain of blockchain (and more generally DLTs), ZKPs experience several challenges that range from enabling adoption and ensuring system maintainability to threat mitigation. From an adoption perspective, the key challenge is ensuring that ZKPs do not diminish user experience. This may require optimizing execution times (since ZKPs may slow down the DLT network or make it unresponsive) and minimizing proof length to help manage the ledger growth (Zhou, 2024). From a maintainability viewpoint, ZKP faces the challenge of balancing user privacy while complying with regulatory requirements that enable lawful access when necessary. Another challenge faced by ZKPs in the domain of blockchain is the need for establishing standards and industry-wide protocols that enable data exchange, helping to ensure interoperability between various legacy systems and ZKP-enabled blockchains. Vulnerability mitigation in ZKP-enabled blockchains presents yet another challenge that must be addressed to prevent catastrophic system failure that may render the DLT itself useless (e.g., by storing invalid or weak proofs). In order to address this challenge, improvements on the integration of ZKP algorithms must be devised to ensure technology-specific vulnerabilities do not result in system-wide compromises.

Lastly, despite the number of potential benefits and optimizations designed to improve their performance, implementation barriers still prevent the widespread use of ZKP in resource-

[1] The subtle differences in time execution complexity are discussed in Appendix B.1

constrained environments (e.g. the typical Internet of Things (IoT) device). This is in large part due to the limited number of cryptographic functionalities that these environments can efficiently support (Chen, 2023). Clearly such systems benefit from optimizations, but incorrectly applied cryptographic optimizations could (and have) resulted in system compromises (Chen, 2023). In addition, ZKPs, like many other security mechanisms, operate better when used in a layered architecture (e.g., there is a security perimeter), a feature that is not always possible in resource constrained systems. This may leave ZKPs vulnerable to local malware or backdoors that could potentially tamper with the proofs being generated.

3.3 Computational Overhead of Zero-Knowledge Proofs

Partala et al. compiled an in depth-review on the performance and computational characteristics of some of the most commonly used ZKP schemes. The article included preprocessing (if applicable), proof length, and the prover and verifier complexities (Partala, 2020). The authors organized their research into five main families of ZKP that covered methods based on 1) Probabilistically Checkable Proofs (PCP); 2) Discrete Logarithm Problem (DLP); 3) Quadratic Arithmetic Programs (QAP); 4) Proofs for Muggles approaches; and 5) Multiparty Computation (MPC). The evaluations performed in the articles were typically based on simple computations (e.g., computation of a cryptographic hash function). Each ZKP scheme evaluated required that computations be converted into their circuit representations. Although manual circuit generation tools can be used to assemble application-specific optimizations, the paper preferred the use of automatic or default circuit generators to ensure fairness. The circuit generation tool and the size of the resulting circuit has a large impact on the efficiency of the schemes, which partially accounts for differences in the circuits among the different schemes. In some cases (e.g., arithmetic and Boolean), it is impossible for the circuits to be fixed. Other aspects that have an impact on the performance of the ZKP scheme are the number of operations that can be performed in parallel, the computation's input size, the security level required, whether the prover and verifier algorithms are optimized, and the nature of the computation.

The ZKP schemes based on PCP include:

- SCI (Scalable Computational Integrity): An interactive computational integrity protocol that does not require a trusted setup to provide publicly verifiable proofs but does not provide complete zero knowledge.

- STARK (Zero-Knowledge Scalable Transparent Argument of Knowledge): A post-quantum secure, interactive scheme which can be made non-interactive through the Fiat-Shamir paradigm. The security of this scheme is dependent on non-standard cryptographic assumptions relating to Reed-Solomon codes.

- Aurora: A succinct, non-interactive argument scheme based on STARK that is designed for rank-1 constraint systems[1]. It is considered post-quantum secure.

- Fractal – A pre-processing SNARK (Succinct Non-Interactive Argument of Knowledge) whose setup uses public randomness and whose proofs can be recursively composed.

[1] In Rank-1 Constraint Systems (R1CS), each constraint is expressed as a rank-1 matrix equation. Intuitively this means that a polynomial such as $x_1^2 + x_1 + 1$ can be expressed using simple operations that only involve two terms at a time ($u = x_1 * x_1;\ v = u * x_2;\ y = x_1 + 1; z = v + y$), which itself can be represented by a binary tree. A more formal definition can be found in (Tari Labs university 2024)

The ZKP schemes using the “proofs-for-muggles” approach are:

- CMT: The first practical implementation of the “proofs-for-muggles” method (Partala, 2020).
- Hyrax: A method that does not require a trusted setup and is based on interactive proofs and cryptographic commitment schemes. It is not post-quantum secure due to its construction being based on the discrete logarithm problem.
- Libra: An implementation that requires a trusted setup whose complexity is dependent on the circuits input. It is not post-quantum secure because the schemes utilize bilinear pairing and knowledge-of-exponent assumptions.
- Spartan : A method that requires no trusted setup and uses polynomial commitments to improve the complexity of verification.

The ZKP schemes based on DLP include:

- Groth’s Linear SNARK is a ZKP scheme introduced by Groth in (Groth, 2009) that has a communication complexity that is proportional to the square root of the circuit’s size.
- BCCGP is a NZKP based on the discrete logarithm problem and Groth’s techniques.
- Bulletproofs is a method based on BCCGP and provides communication-efficient proofs for performing transactions confidentially. The scheme can be made quantum secure through the Fiat-Shamir heuristic.
- Groth16 is a ZKP scheme that utilizes elliptic curve pairings and knowledge-of-exponent to achieve “perfect completeness and zero-knowledge with computational soundness” (Partala, 2020).
- Sonic utilizes a polynomial commitment scheme, pairings, and arithmetic circuits. PLONK is an open-source implementation written in Rust that improves the efficiency of the prover algorithm.
- Marlin is a ZKP method based on Sonic that improves the verification time complexity through the implementation of special encoding for the statement. An open-source implementation written in Rust is available.
- Supersonic improves upon the Sonic scheme by introducing a polynomial commitment scheme to remove the need for a trusted setup.

The ZKP schemes based on QAP are:

- GGPR: The first ZKP scheme to be applied to QSP or QAP computations.
- Pinocchio: An implementation based on GGPR that provides the capability for anyone to verify proofs.

The ZKP schemes from MPC include ZKGC, ZKBoo, ZKB++, and Ligero. ZKGC is based on Yao’s garbled circuits and has a proof-of-concept implementation. ZKBoo is built on the “MPC-in-the-head” approach and provides quick proving and verification processes, at the cost of

large proof lengths. It also has a proof-of-concept implementation. ZKB++ has a practical implementation and improves upon ZKBoo by reducing the proof length by 50%. Ligero is based on symmetric cryptography, MPC, and PCPs. It is considered post-quantum secure. A graphical taxonomy of the aforementioned methods, including some of their core capabilities is documented in Figure 13.

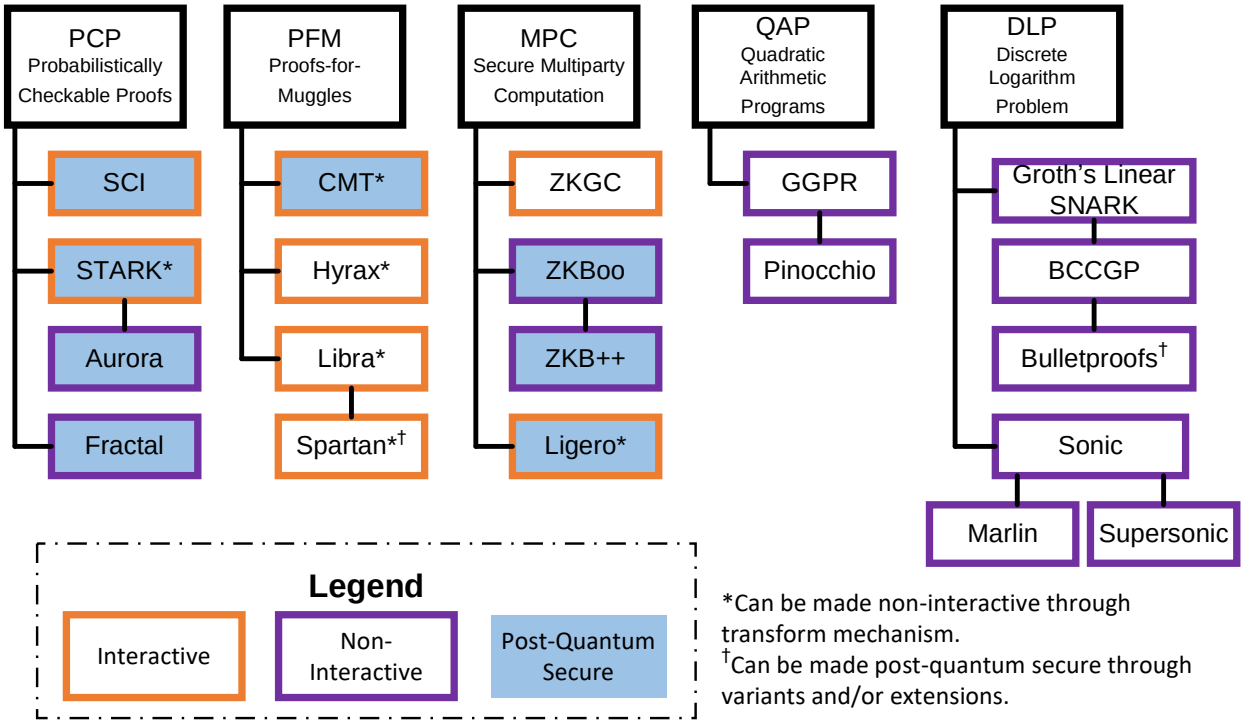
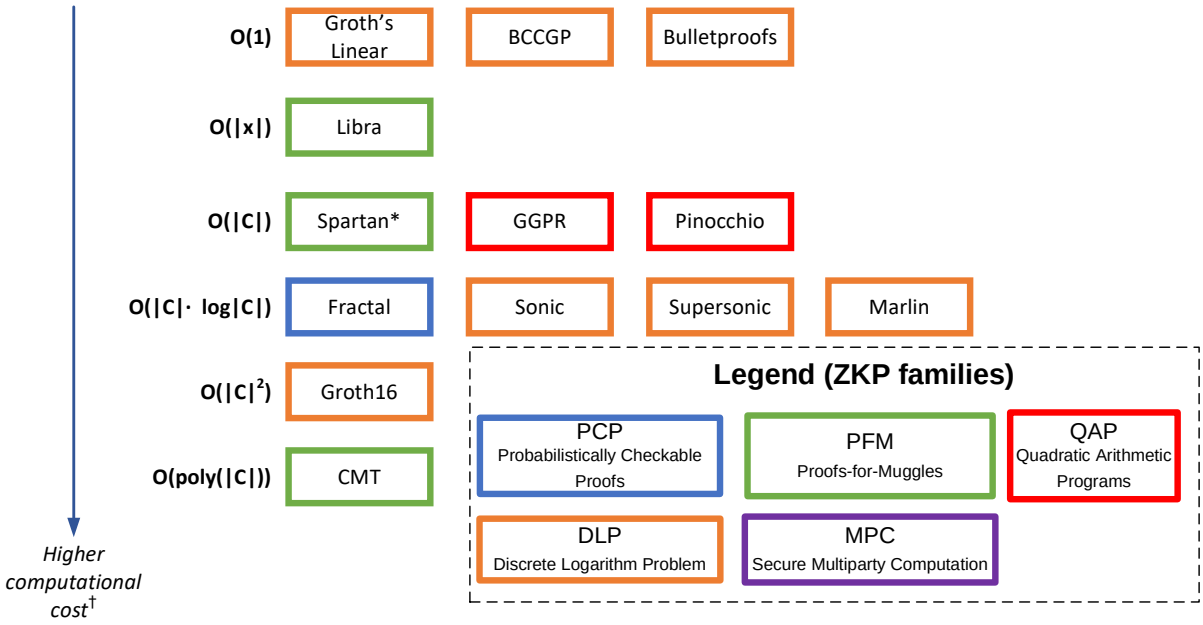


Figure 13: A taxonomy of ZKPs, grouped by their principle of operation. Note that some implementations may be based on one or more primary ZKPs but have been grouped according to their dominant source.

Figure 14 through -Figure 17 illustrate the time complexities of various operations performed by the ZKP schemes. Within the figures, $|C|$ is the number of gates in the circuit C, x is the input to the circuit, w is the witness, M is the number of multiplication (AND) gates in the circuit, d is the depth of the circuit C, G is the width of the circuit C, $s(|x|)$ is the amount of memory taken by the computation, and N is the length of inputs and outputs of the computation.



†Variables such as C and x may modify the order of computational cost.
*One time public computation by the verifier. Setup is not repeated for different inputs.

Figure 14: Preprocessing Time Complexities of Applicable ZKP Schemes

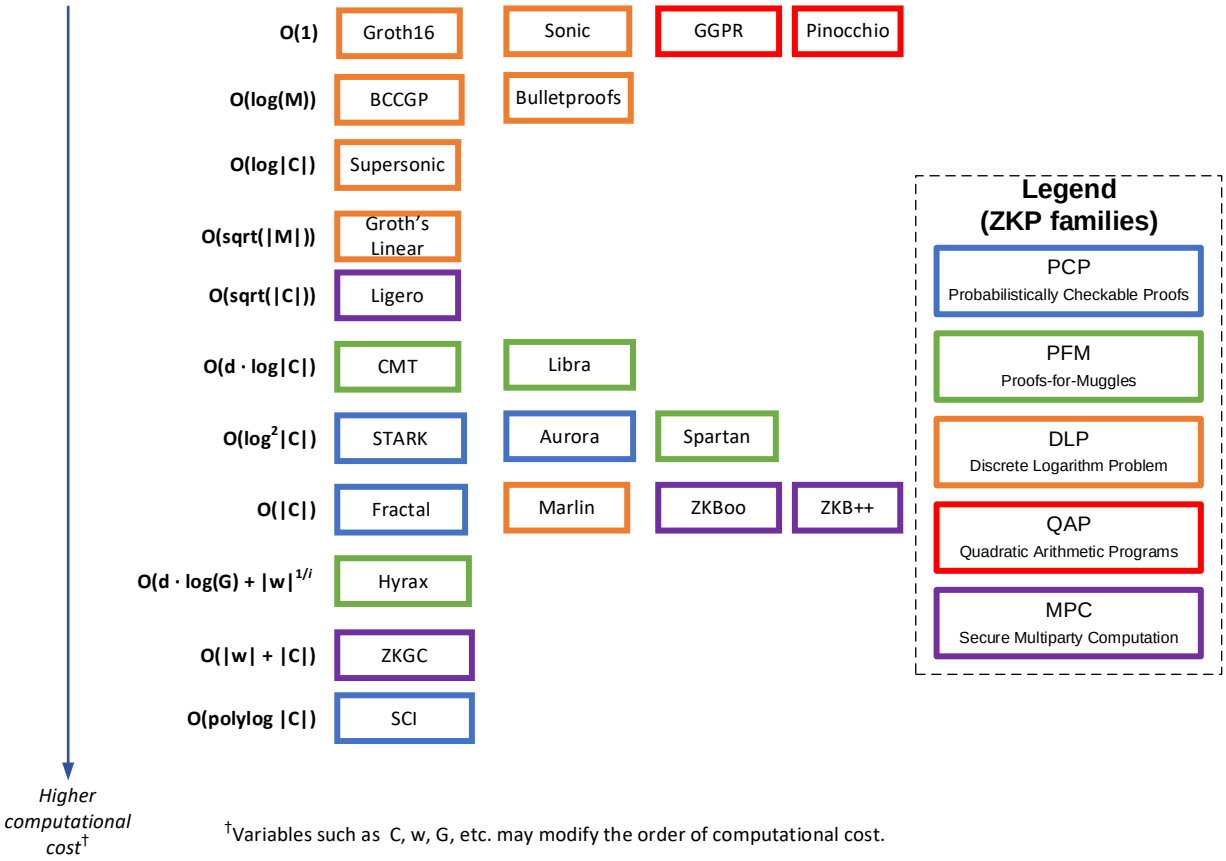


Figure 15: Taxonomy of Communication/Proof Lengths of ZKP Schemes

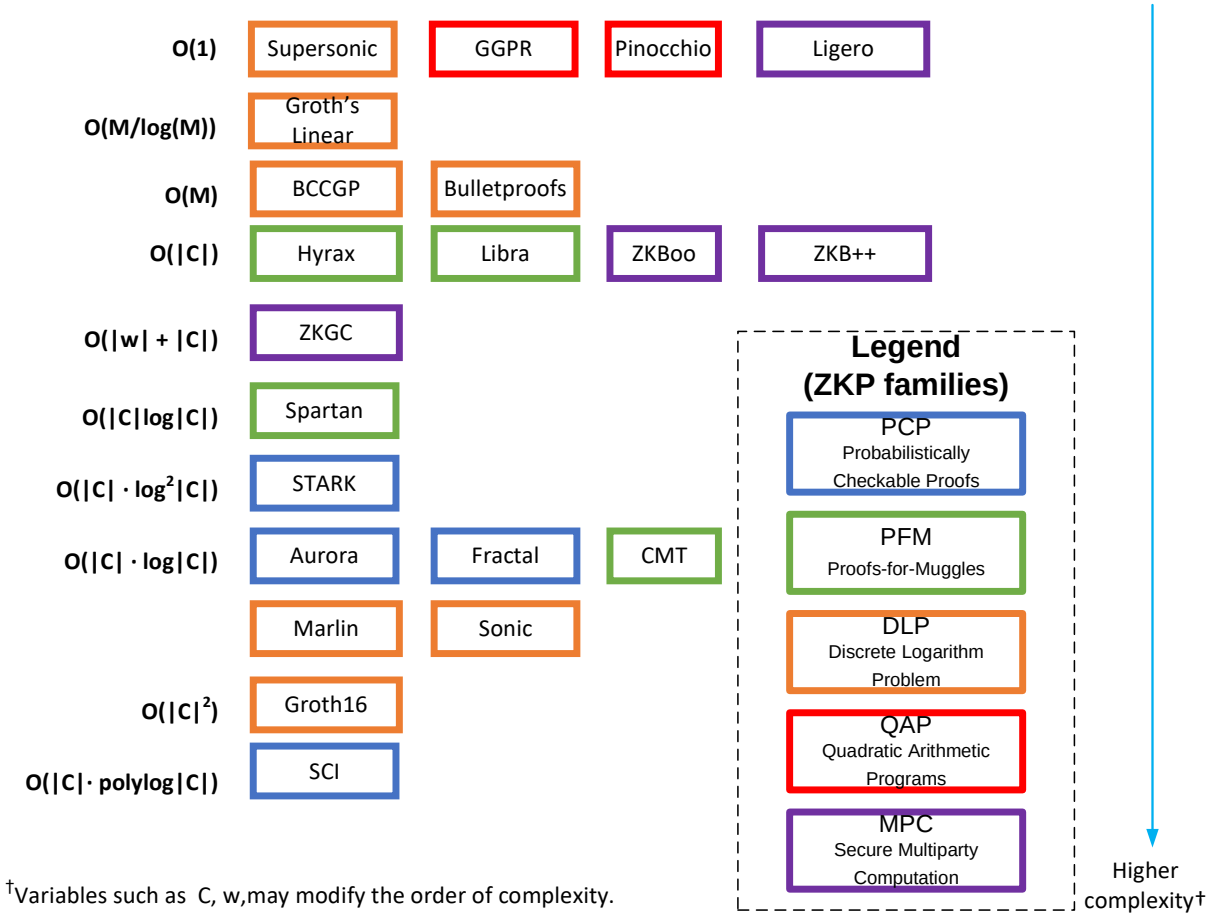


Figure 16: Time Complexities of ***Prover*** Algorithms in ZKP Schemes

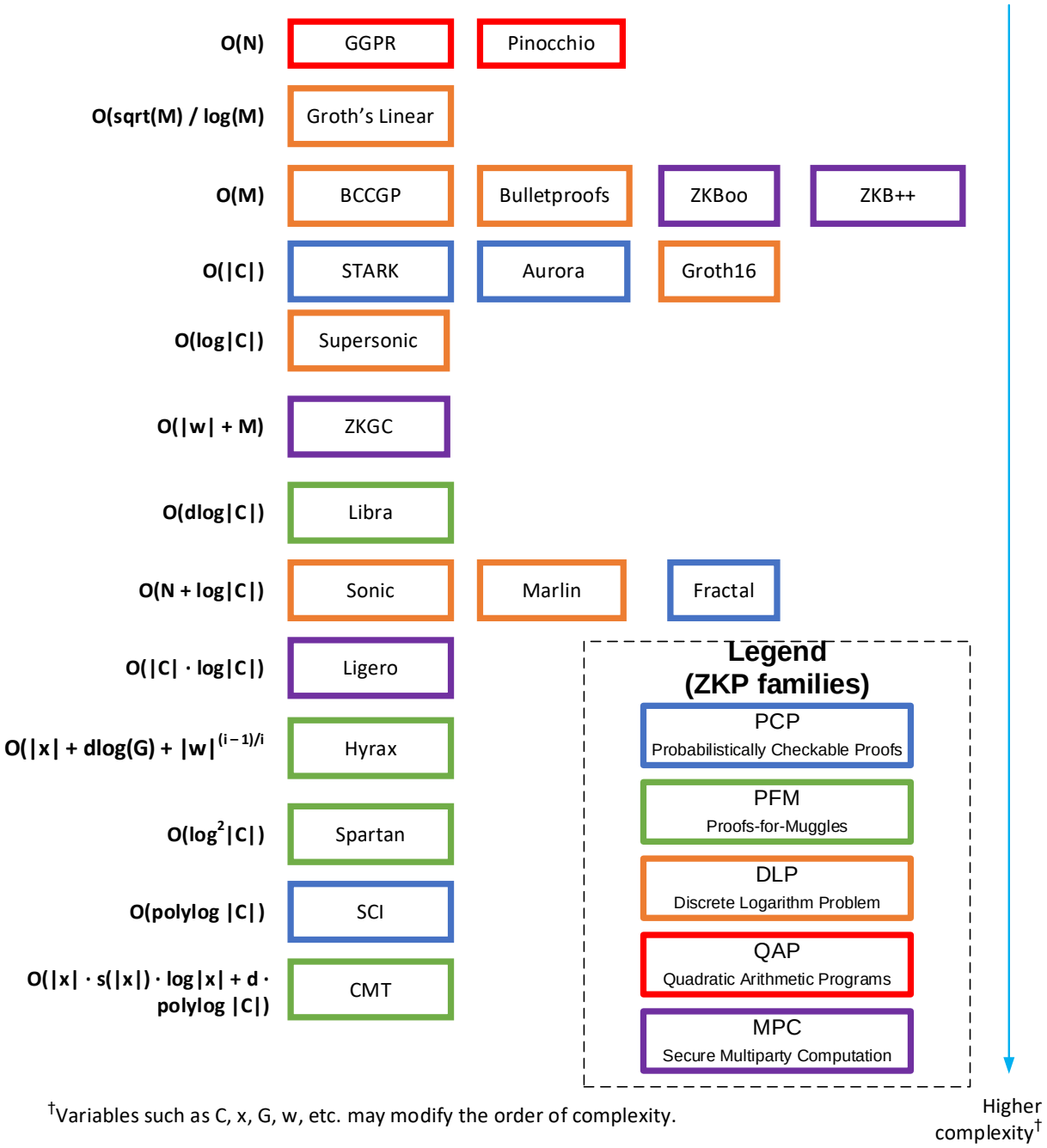


Figure 17: Time Complexities of **Verifier** Algorithms for ZKP Schemes

As illustrated by Figure 14-Figure 17, there is no ZKP scheme that performs all operations better than the rest. For example, Sonic has constant communication and proof lengths but has $O(|C| \cdot log|C|)$ time complexity for its prover algorithm. Hyrax has a better time complexity than Sonic for its prover algorithm ($O(|C|)$), but a worse time complexity ($O(d \cdot log(G) + |w|^{1/i})$) for its communication and proof lengths. For algebraic statements, interactive ZKPs are more

efficient than NZKP's, while they are outperformed by NZKP's when computations of block ciphers or hash functions are required (Partala, 2020). Ligero works best for computations done sequentially, while Hyrax performs best on parallel computations. QAP generally performs better than QSP on practical computations.

4.0 Differential Privacy

Differential Privacy (DP) is an approach to formulating privacy goals where the risk to an individual's privacy "should not substantially increase as a result of participating in a statistical database" (Dwork, Differential privacy, 2006). This is accomplished by providing relative guarantees about data disclosures, rather than absolute guarantees. With DP, any given disclosure will not change significantly whether the individual's data is included in the dataset or not, hence ensuring that a user's participation (or lack of) cannot be inferred by an external party[1]. A DP algorithm is said to achieve ε-DP, where ε is called the privacy budget, if and only if the level of privacy provided by the algorithm satisfies the following equation:

$$Pr[A(D_1)\ \epsilon\ S]\ \leq e^{\varepsilon} \times Pr[A(D_2\ \epsilon\ S)] \quad (Eq.1)$$

Where A is a randomization algorithm, D_1 and D_2 are adjacent databases (e.g., a dataset with and without an individual), and S is *all* the possible outputs of A. Due to its strict mathematical requirement, it might be computationally expensive to obtain ε -DP (Jiang H. J., 2020) and thus many practical implementations of DP have used an alternative approach which can be defined as:

$$Pr[A(D_1)\ \epsilon\ S]\ \leq e^{\varepsilon} \times Pr[A(D_2\ \epsilon\ S)] + \delta \quad (Eq.2)$$

The equation shown in Eq. 2 is known as (ε, δ)-DP, and offers a relaxation mechanism that can be used to approximate the behavior of an ideal ε -DP algorithm. The numerical parameter δ controls the probability of not satisfying ε -DP, while most of the time (with probability $1 - \delta$) it ensures data remains private. Thus, by varying the δ parameter it becomes possible for application developers to tune the privacy parameters in terms of accuracy and computational efficiency. A sensitivity value[2], determined by the largest change that results from adding or removing a user's data from the dataset, determines the amount of noise applied to the data. The amount of noise that must be applied to the dataset is directly proportional to the sensitivity value and the privacy loss (Husnoo, 2021).

Various mathematical mechanisms may be used to achieve (ε, δ)-DP or (ε -DP). Some of the most commonly used mechanisms that can be used to shape the noise behavior include the Laplace, Exponential, and Gaussian mechanisms (Husnoo, 2021). The Laplace algorithm can be used when individual samples are highly heterogeneous, or when outliers (e.g., extreme values) are present and must be masked. The exponential algorithm is useful for situations where the output is not a continuous variable (e.g., class bins), or when an actual population sample must be returned instead of a synthetically generated result (Near & Abuah, 2024). The Gaussian mechanism simplifies understanding of the statistical properties of datasets, potentially enabling researchers to combine results from multiple queries due to the additive properties of the Gaussian distribution.

Like many other privacy constructs, DP can be assembled and combined in complex architectures to address application-specific requirements (Jiang H. J., 2020). These DP

[1] This property only holds true if the population samples are independent from each other (e.g., all sampled values are independent variables)

[2] Sensitivity can be formally defined as $\Delta f = \max_{D_1, D_2} \| f(D_1) - f(D_2) \|$ for $f: P \rightarrow R^k$. In particular, when $k = 1$ the sensitivity of f is the maximum difference in the values that the function f may result from a pair of profiles that differ by only one record or unit count.

algorithms may be applied sequentially on a single dataset or in parallel among separate datasets. When applied sequentially, the level of privacy preservation provided is the sum of each DP-algorithm's privacy budget[1]. When applied in parallel, the overall level of privacy preservation provided is determined by the DP-algorithm with the largest privacy budget.

Although DP mechanisms can be adapted to a wide variety of application contexts, two of the most commonly architectural variations include a) their ability to handle dependent / independent variables; and b) their ability to process (and protect) data in a local or centralized manner (Jiang H. J., 2020). Dependent DP mechanisms are required when datasets contain attributes that are correlated or dependent on each other (e.g., height and weight). Under a standard DP mechanism, the addition, removal, or modification of such tuples could allow a malicious actor to infer sensitive information even when the result appears to be DP-protected. Examples of implementations that can handle dependent datasets, include (Liu C. S., 2016), (Zhao, 2017), and (Almadhoun, 2020).

Local DP processing mechanisms are the preferred solution for scenarios where data is collected by an untrusted third-party. This is in stark contrast to global DP schemes, where noise is added to the data after it has been sent and curated by a trusted server (Wang T. X., 2020). In a local DP mechanism, the responsibility of adding noise is assigned to the field agent, who applies it to their data stream before forwarding it to the data collector. The random response technique is the primary method of perturbing the data in local DP that essentially introduces a high-level of uncertainty (e.g., noise), which cannot be reversed by an external agent (Jiang H. J., 2020). Although the concept is relatively simple, the amount of noise that the local agent must introduce is often based on the worst-case scenario, an assumption that may limit data accuracy, potentially rendering the data useless for future analysis[2]. A simple example of this method is described in Appendix B.4.

4.1 Benefits and Applications of Differential Privacy

From a functional perspective, DP can ensure protection against a wide variety of de-anonymization attacks, hence preventing individuals from being re-identified in most cases (Husnoo, 2021). However, from a regulatory perspective, the benefit of DP is that the amount of privacy provided to an individual can be measured and controlled. This enables numerical comparisons to be performed among distinct privacy preserving implementations, potentially enabling benchmarking to be a decisive factor in technology adoption.

From a data processing perspective, DP has been used in statistical estimations, data publishing, data mining, and machine learning applications (Jiang H. J., 2020). Its data-agnostic approach enables DP to be implemented to address a wide range of challenges associated with the energy, transportation, and healthcare domains (Husnoo, 2021). In the energy domain, the researchers in (Alisic, 2020) determined that a malicious actor could infer information about a change in occupancy of a smart home through the sensors placed in the home (Husnoo, 2021).

[1] A privacy budget is a value defined by the data curator based on the application's privacy sensitivity. It defines the maximum amount of privacy loss that is permissible while generating or accessing the protected dataset (when interactive querying mechanisms are used).

[2] Note that under certain applications, the amount of noise being introduced can exceed the amount of noise that a particular application can tolerate. This is by no means a failure of DP, rather it's a mismatch between the target privacy goal, the dataset characteristics and the mechanism employed to provide DP (Dwork and McSherry, Differential Privacy – A Primer for the Perplexed 2011).

To eliminate this risk, the authors developed a DP scheme that utilized Gaussian noise to obscure energy patterns that could be used to predict home occupancy. In (Hossain, 2021), the researchers developed a DP scheme that generated Laplacian noise by oscillating the charge state of rechargeable batteries used for demand-side energy management, effectively protecting their true state while giving others the opportunity to have confidence on the overall capacity. In (Gai, 2022), the researchers developed a differentially private data aggregation scheme where noise could be added to smart meter data through randomized response. The proposed scheme does not require a trusted third party and the results of a performance analysis determined it minimized computation and communication overhead.

Within the transportation domain, researchers in (Jiang K. D.-L., 2013) developed a DP-based sampling distance and direction technique to preserve the privacy of ship trajectories (Husnoo, 2021). Their analysis of their method showed that it accomplished a good balance between privacy and utility while remaining usable when compared to other methods of injecting noise into data. In (Qiu, 2021), the researchers proposed a DP dynamic data stream publishing mechanism that implemented adaptive sampling, variable windows, privacy budget allocation, packet perturbation, and filtering mechanisms to preserve the privacy of EV data (e.g., location, driver identity).

In the healthcare domain, the researchers in (Guo J. M., 2021) proposed a data publishing method based on $k - anonymity$ and a temporal DP mechanism to prevent the leakage of private information (e.g., heartrate, blood oxygen, etc.) from IoT wearable devices. The researchers in (Mohammed, 2015) demonstrated a data management framework for data mining that utilized encryption and a DP query interface to preserve the privacy of patient data. The framework was evaluated with a publicly available breast cancer dataset consisting of 286 records with 9 attributes. To further summarize the different types of research, a review of DP-based solutions is described in Table 9.

Table 9: Applications of DP Across Various Domains

Citation	Keywords	Overview
(Chamikara, 2020)	Face Recognition, Machine Learning, Local DP	A privacy-preserving face recognition scheme called Privacy using EigEnface Perturbation (PEEP) was developed by the researchers using local DP. The scheme generates eigenfaces of the original image and applies Laplacian noise to randomize the image. The noisy image is sent to a third-party server and a machine learning model is trained on the perturbed data. The performance of their scheme was evaluated on the open face image dataset that is available from the University of Massachusetts website and the large-scale Celeb-Faces Attributes (CelebA) dataset. 70% of the data was used for training and 30% for testing. The results showed that the scheme achieved a classification accuracy of 70%-90% using standard privacy settings.
(Yin, 2021)	Federated Learning, Multiparty data sharing, Functional Encryption	A federated learning method utilizing Bayesian DP was developed to protect an individual's privacy from being inferred from the training model parameters (e.g., model weights). The method used function hiding, multi-input function encryption to obscure the model parameters sent to the server. The method was evaluated using a convolutional

		neural network (CNN) and the MINST dataset splitting method. The experiments showed that Bayesian DP can achieve a classification accuracy of 91.1%.
(Jiang B. J., 2021)	Smart Home, Internet of Things, Secure Routing, Energy Efficiency	The researchers developed an efficient and privacy-preserving traffic obfuscation (EPIC) framework to mitigate traffic analysis attacks that could reveal sensitive information about the occupant(s) such as sleep patterns and medical conditions. It guarantees strong DP through the use of a network of connected smart homes. A smart home's traffic is sent to another smart home's gateway (to act as a proxy gateway) before it is transmitted to the Internet. A directed random walk (DRW) scheme was used for uploads and a combination of DRW and flooding were used for downloads. The framework was evaluated on simulations based on a community in Gainesville, Florida, USA that is composed of 77 total homes in a 640 x 300 m^2 area. The results showed that their scheme outperformed other DP mechanisms in preserving privacy and lowering network energy consumption.
(Hassan, 2019)	Smart Grid, Renewable Energy Resources	A DP-based real time load monitoring (DPLM) approach was proposed by the researchers to preserve the privacy of the user's routines and usage of specific renewable energy resources. The scheme first calculates the sum, S, of the energy readings from all energy sources for a given timeframe. If S is greater than 0, then: $$SN = S + N + EX$$ Where N is the value of added Laplacian noise and EX is an extra value. SN is compared with a selected maximum peak load value P, and if SN is greater than P, the value of SN is set to P and transmitted to the utility. The value of $SN - P$ is stored in EX and used for the calculations in the nest timeframe. Their scheme was evaluated on real datasets composed of grid energy consumption data from the Residential Energy Consumption Survey for the Midwest, and solar and wind energy from the Hong Kong Observatory. The data was iterated over 31 days with a transmission interval of 10 minutes. The monthly output results showed that their scheme achieved an error rate of only 1.5% when P was 1,200 Wh. When P was 1,200 Wh, the total energy reported was 4,844 kWh and the masked value was 4,916 kWh.
(Eibl, 2018)	Smart Meter, Load Forecasting	The researchers designed a DP scheme for load forecasting while preserving the privacy of electrical production and consumption data of homes. DP is guaranteed by grouping households into zones (to avoid requiring individual load data and applying the Laplace mechanism to each zone's aggregated load data. HE or masking protocols are used to perform the summation of all the household's data in the zone. The scheme was evaluated against three forecasting methods and found that each household had an individual membership inference risk of less than 60% and only 10% over random guessing.
(Ju, 2023)	Electric Vehicle, Local DP, Data	The researchers proposed a privacy-preserving data range query (PPQ) scheme based on local DP to protect sensitive EV data (e.g., location, charging time, remaining power). The

	Range Query, Vehicle-to-Grid	scheme is able to resist collusion attacks by removing the need for a trusted third party. Additional data protection is provided through a data encryption optimization model that enables EVs to locally add noise to their data. The scheme was evaluated in a vehicle-to-grid simulation over simulated datasets (random, normal, Laplace, Zipf) and a real-world dataset (EVs – One Electric Vehicle Dataset). The results showed that the scheme could successfully balance efficiency with accuracy while ensuring data privacy.
(Parker, Hale, & Barooah, 2022)	Smart meter time series, Spectral Differential Privacy (SpDP)	The authors propose a time-series oriented DP mechanism called SpDP. The method aims to protect a signal's power spectral density (PSD), using a frequency-domain approach that prevents extracting event signatures typically found in demand profiles. Protections can be applied at the edge, further increasing its privacy attributes. A key contribution of the work is that SpDP guarantees a minimum level of privacy (to the PSD) while being independent of the time-series duration.
(Ravi, et al., 2022)	DP-enabled K means clustering, Smart meter time series	The researchers developed a K-means clustering method that enable data aggregators to generate representative load-shape data profiles while maintaining privacy. This is done by using gaussian noise to generate cluster centroids that accurately reflect the load-shape behavior of users without the risk of releasing an actual demand curve. The proposed algorithm could be used to assemble differentially private synthetic load patterns that could be used to obtain summary statistics or create labeled datasets.

4.2 Challenges Faced by Differential Privacy

A common challenge faced during the implementation of a DP-based framework is determining the ideal ε -value which defines the balance between utility and privacy (Husnoo, 2021). If a small ε -value is selected, more noise is added to the data, helping to increase its privacy at the cost of a lower accuracy, which in turn may decrease its utility. However, if a large ε-value is chosen, less noise is applied to the data which increases the risk that the privacy will be compromised while increasing the overall accuracy (and thus its utility). There are currently neither standard guidelines nor methods for choosing the ideal ε-value, hence this value must be determined through experimentation[1]. A similar challenge is faced with determining the sensitivity value of the algorithm. Lower sensitivity values provide increased data utility while severely compromising its privacy. Likewise, higher sensitivity values increase the privacy of the data while reducing data utility. Optimal values have only been found for specific types of datasets, but most practical solutions have to self-determine the maximum amount privacy loss that is tolerable for their particular use-case. Another challenge found in DP-based solutions is that some distributions (e.g., Laplace) may lead to DP-protected datasets that exhibit different statistical properties (bias and variability), even if they share the same initial parameters.

[1] See https://desfontain.es/blog/real-world-differential-privacy.html for a sample list of ε values used in end-user applications. A more formal discussion is presented in (Dwork, Kohli and Mulligan, Differential privacy in practice: Expose your epsilons! 2019)

A large limitation of DP is that it cannot guarantee complete privacy preservation (Husnoo, 2021). An example of when DP may fail is when datasets or individual records within have a high correlation with each other. A malicious actor can use these correlations to make inferences about individuals in the datasets and compromise their privacy. In fact, in (Haeberlen, 2011), the researchers exposed vulnerabilities in well-known implementations of DP that could be used to leak private information through covert channel attacks. Transformation-based methods have been used to address this issue; however, they currently only work in specific cases and may compromise data utility in other cases.

4.3 Computational Overhead of Differential Privacy

The standard local DP protocols for frequency estimation on categorical data (i.e., event counting) are based on direct perturbation, unary encoding, hash encoding, transformation, and subset selection (Wang T. X., 2020). Direct perturbation applies the noise to the data directly through randomization. Binary randomized response (BRR) is applicable to situations where there are only two possible answers to a query. Generalized randomized response (GRR) is utilized when there are more than two possible responses to a query. Encoding the original value first into a vector and then applying noise to each bit in the vector is called unary encoding (UE). Hash encoding follows the same process as UE but provides randomization differently. In basic Randomized Aggregable Privacy-Preserving Ordinal Response (RAPPOR), a permanent randomized response is used instead of the real response and an instantaneous randomized response reports on the permanent randomized response over time until it's revealed (Erlingsson, 2014). The transformation method is generally used to lower the cost of communication (Wang T. X., 2020). For example, S-Hist was developed in (Bassily, 2015) to create a histogram of the most frequent values. The communication cost was lowered by choosing a random bit from the vector generated from encoding the original value based on random matrix projection. In subset selection, ω is the size of the subset that is randomly selected from the total set of items. Table 10 describes the communication costs, error bounds, and variances of various algorithms used for frequency estimation.

Table 10: Communication Cost, Error Bound, and Variance of Various Local DP Algorithms From (Wang T. X., 2020)

Encoding Method	Local DP Algorithm	Communication Cost	Error Bound	Variance	Know Domain?
Direct Perturbation	BRR	O(1)	$O\left(\frac{1}{\epsilon\sqrt{N}}\right)$	$\frac{e^{\epsilon}}{N(e^{\epsilon}-1)^2}$	Y
	GRR	O(log k)	$O\left(\frac{\sqrt{klog(k)}}{\epsilon\sqrt{N}}\right)$	$\frac{e^{\epsilon}+k-2}{N(e^{\epsilon}-1)^2}$	Y
Unary Encoding	SUE	O(k)	$O\left(\frac{\sqrt{log(k)}}{\epsilon\sqrt{N}}\right)$	$\frac{e^{\epsilon/2}}{N(e^{\epsilon/2}-1)^2}$	Y
	OUE	O(k)	$O\left(\frac{\sqrt{log(k)}}{\epsilon\sqrt{N}}\right)$	$\frac{4e^{\epsilon}}{N(e^{\epsilon}-1)^2}$	Y
Hash Encoding	RAPPOR	Θ(k) (user) Θ(Nk) (aggregator)	$O\left(\frac{k}{\epsilon\sqrt{N}}\right)$	$\frac{e^{\epsilon/2}}{N(e^{\epsilon/2}-1)^2}$	Y
	O-RAPPOR	Θ(k)	$O\left(\frac{k}{\epsilon\sqrt{N}}\right)$	$\frac{e^{\epsilon/2}}{N(e^{\epsilon/2}-1)^2}$	N

	O-RR	O(log k)	$O\left(\frac{\sqrt{klog(k)}}{\epsilon\sqrt{N}}\right)$	$\frac{e^{\epsilon}+k-2}{N(e^{\epsilon}-1)^2}$	N
	BLH	O(log k)	$O\left(\frac{\sqrt{log(k)}}{\epsilon\sqrt{N}}\right)$	$\frac{(e^{\epsilon}+1)^2}{N(e^{\epsilon}-1)^2}$	Y
	OLH	O(log k)	$O\left(\frac{\sqrt{log(k)}}{\epsilon\sqrt{N}}\right)$	$\frac{4e^{\epsilon}}{N(e^{\epsilon}-1)^2}$	Y
Transformation	S-Hist	O(log b)	$O\left(\frac{\sqrt{log(k)}}{\epsilon\sqrt{N}}\right)$	$\frac{e^{\epsilon}}{N(e^{\epsilon}-1)^2}$	Y
	HRR	O(log k)	$O\left(\frac{\sqrt{log(k)}}{\epsilon\sqrt{N}}\right)$	$\frac{4e^{\epsilon}}{N(e^{\epsilon}-1)^2}$	Y
Subset Selection	ω-SM	O(ω)	$O\left(\frac{\sqrt{klog(k)}}{\epsilon\sqrt{N}}\right)$	$\frac{e^{\epsilon}+k-2}{N(e^{\epsilon}-1)^2}$	Y

In Table 10, k is the cardinality of the dataset, b is the single bit selected in the S-Hist scheme, ω is the size of the subset of data selected in subset selection methods, ϵ is the privacy budget value, and N is the number of users in the dataset.

For datasets where users have a varying number of elements, various mechanisms are utilized for frequency estimation which include item distribution estimation, frequent items mining, and frequent itemsets mining (Wang T. X., 2020). Item distribution estimation involves analyzing the distributions over some number of items. Frequent item mining discovers all the items that occur at least some number, ω, times in the dataset. Frequent itemsets mining is similar to frequent item mining, the difference being that it works on itemsets rather than individual items. New terms discovery is applied to situations where item domain is unknown (e.g., finding the most used terms from varying numbers of users with varying responses).

Table 11: Communication Cost of Various Local DP Algorithms on Set-Valued Data From (Wang T. X., 2020)

Task	LDP Algorithm	Communication Cost	Know Domain?
Item distribution estimation	PrivSet	$O(l')$	Y
	LDPart	$O(\|V\|_m)$	Y
Frequent item mining	TreeHist	$O(1)$	Y
	LDPMiner	$O(logk+\omega)$	Y
	PEM	$O(logk)$	Y
	Calibrate	$O(k)$	Y
Frequent itemset mining	Personalized	$O(k)$	Y
	SVSM	$O(logk)$	Y
New terms discovering	A-RAPPOR	$O(logk)$	N
	PrivTrie	$O(\|V\|_m)$	N

In Table 11, l' is the output size of randomization, $|V|_m$ is the maximum number of nodes in a partition tree that partitions users into different groups, k is the total number of items in the domain across all users, and ω is an integer value describing the minimum number of times an item has occurred in the items or itemset (e.g., $\omega = 7$, all items with at least 7 occurrences).

For computing k-way marginal probability distributions, RAPPOR and several other schemes have been developed. As the number of dimensions k increases, RAPPOR suffers a large drop in efficiency and accuracy. Fanti et al. proposed an expectation maximization (EM) based algorithm in (Fanti, 2015) that is limited to 2-way marginals but suffers from high time and space overheads as k grows larger. In (Ren, 2018), the researchers developed a local DP scheme using a Lasso-based regression mechanism and dimension reduction through discovering compactly correlated attributes to reduce time overhead and increase data utility compared to earlier schemes. The researchers in (Wang T. X., 2019) proposed a DP scheme that used Copula theory to resolve the issue of exponential growth as the dimension k grew in size by reducing the number of estimations from k to one and two-marginal distributions. In (Cormode, 2018), the researchers developed a local DP scheme that utilized the Hadamard transform technique to improve accuracy and reduce communication costs. However, non-binary attributes must be transformed into binary types, which adds more dimensions to the data. In (Zhang Z. T., 2018), the researchers developed a consistent adaptive local marginal (CALM) algorithm "that builds k-way marginals by taking the form of m marginals each of size l". Table 12 provides a summary of the communication costs, variance, time complexity, and space complexity of previously discussed local DP schemes.

Table 12: Communication Cost, Variance, Time Complexity, and Space Complexity of LDP Algorithms From (Wang T. X., 2020)

LDP Algorithm	Communication Cost	Variance	Time Complexity	Space Complexity
RAPPOR	$O\left(\prod_{j=1}^{d}\lvert\Omega_j\rvert\right)$	$2d * Var$	High	High
(Fanti, 2015)	$O\left(\sum_{j=1}^{d}\lvert\Omega_j\rvert\right)$	$2d * Var$	$O\left(N * \sum_{i=1}^{k}\binom{d}{i} * 2^i\right)$	$O\left(\sum_{i=1}^{k}\binom{d}{i} * 2^i\right)$
LoPub	$O\left(\sum_{j=1}^{d}\lvert\Omega_j\rvert\right)$	$2d * Var$	Medium	High
(Cormode, 2018)	$O\left(\sum_{i=1}^{k}\binom{d}{i}\right)$	$\sum_{i=1}^{k}\binom{d}{i} * Var$	$O\left(N + \binom{d}{k} * 2^k\right)$	$O\left(\sum_{i=1}^{k}\binom{d}{i}\right)$
LoCop	$O\left(\sum_{j=1}^{d}\lvert\Omega_j\rvert\right)$	$2^d * Var$	Low	High
CALM	$O(2^l)$	$\frac{m}{N} * 2^l * Var$	$O(N * 2^l)$	$O(m * 2^l)$

In Table 12, Var is the variance of estimating a single cell in the full table, l is the size of m low marginals of the dataset, d is the total number of attributes in the dataset, $\lvert\Omega_j\rvert$ is the cardinality of the attribute in the dataset and N is the total number of users in the data.

As mentioned during this section's introduction, the exponential mechanism is one of the most commonly used methods when a DP query must return a representative sample from a database while maintaining privacy. Within the energy space, this problem could represent releasing a consumption curve that characterizes a consumer's daily demand by picking one that has a certain *resemblance* to the consumer's true demand curve. The exponential mechanism was initially defined by (McSherry & Talwar, 2008), and it states that if the loss of outputting the statistic X is $\ell(\mathrm{X})$, the mechanism will output $\mathrm{x} \in \mathrm{X}$ with probability/density proportional to $\mathrm{e}^{-\mathrm{c}\ell(\mathrm{x})}$, where c is a constant that depends on ϵ (i.e., the sensitivity) (Ganesh, 2022). Despite its relatively simple approach, its main challenge is that it requires computing

$\ell(X)$ for every $x \in X$, and for every possible statistic that may be analyzed in the future. For example, if there 1000 customers within a feeder (denoted by n), and there are 20 metrics (m) of interest (e.g., net consumption, peak load, base load, etc.), then the exponential mechanism must determine $\ell(X)$ over the $m * n$ space.

5.0 Federated Learning

Federated learning (FL) is a machine learning framework introduced by Google in 2015 that enables the training of a *global model* by relying on an *intermediate model* that is used instead of the user's raw data (Banabilah, 2022). This contrasts with other, more standard ML methods, where the user's raw data must be shared with a centralized model creator (e.g., where the model weights are trained). Traditional approaches are known to increase the attack surface because an adversary could potentially obtain access to the sensitive information while the data is in motion or at rest. Even if there are sufficient access control mechanisms, the model creator may abuse the data or unintentionally release it as part of the trained model itself (for example a large language model may output text fragments that plagiarize published works).

Although the implementation details are highly variable, most FL approaches use an iterative training algorithm that gradually reconciles a *global model* with multiple, end-user models that are trained at the edge. This can be done, for example, by first sharing an initial *global model* with the users in the network. The users then re-train the *global model* using their local data and send the updated parameters back to the central server. Then the server aggregates the received parameters, updates the parameters of the global model, and sends the new *global model* to users who can repeat the process until the model converges. Since the raw data never leaves the user's local environment, greater amounts of privacy and security can be achieved. It is important to note that FL is a concept, and it must be complemented with other privacy-preserving mechanisms in order to help ensure the privacy and security of the users' data and the global model itself. Broadly speaking FLs, can be categorized into three main classes a) vertical FL (VFL); b) horizontal FL (HFL); and c) federated transfer learning (FTL) (Banabilah, 2022).

VFL is applicable when there are two datasets that contain a large overlap of users while only having a few features in common (Banabilah, 2022). For example, a bank whose datasets contain information about user income and credit scores can collaborate with an e-commerce website whose datasets contain information about user browsing and purchase history to train a model on the behaviors of their users (assuming users can be linked by their credit card information). The main focus of VFL is preserving privacy. A VFL risk model assumes the presence of honest but curious users who collaborate towards the model creation, but which may try to infer or extract training data during the process.

In contrast, horizontal FL is suitable for situations where two datasets contain similar features, but don't share many of the same users (Banabilah, 2022). For example, two electric companies in different regions would have different users in their datasets, but similar features exist (e.g., energy usage, payment information). By collaborating, the companies could train a more robust model by increasing the number of data samples (e.g., potentially increasing its accuracy). HFL's primary focus is on security, hence algorithms are intended to prevent participants from learning from each other.

Federated transfer learning is suitable for situations where the two datasets contain little overlap of the users and features (e.g., A commercial flight database combined with a fuel distribution database in a model that predicts flight prices based on fuel prices) (Banabilah, 2022). One dataset (source) generally contains more samples than the second dataset (target). The source trains an initial model on feature extraction, while the target uses the trained model with their own dataset to further tune the model.

5.1 Benefits and Applications of Federated Learning

The primary benefit of FL is that it enables collaboration between a large number of users that must train a *global model* while enabling them to keep their raw, private data within their own control (Zhang C. Y., 2021). By keeping user data locally, a layer of privacy is intrinsically created, which may encourage other users to share their data and improve the model (E.g., an increased number of available datasets or records could result in a higher accuracy model). Another benefit of this property is that the central server can reduce its computational requirements because the training data is being pre-processed at the user's local device (this is especially true as the number of users increases) (Banabilah, 2022). Since the central server does not store large amounts of user data, the cost to the service provider is also lowered by eliminating the need to purchase and install high-capacity storage. Another benefit of using FL-based techniques is that their model aggregation properties enable the data processor to comply with privacy laws and regulations (e.g., GDPR) (Zhang C. Y., 2021).

FL has found applications across multiple domains, that span the healthcare, transportation, finance, and energy fields. In the healthcare domain, FL has been used to securely aggregate medical data from various hospitals into a large, *global model* while preserving the patient's data privacy and adhering to emerging privacy laws (Zhang C. Y., 2021). In another example, the researchers in (Lu, 2020) proposed a FL framework that is applicable to the transportation domain. In the study, the researchers aimed to mitigate data leakage in vehicular cyber-physical systems that, if left unprotected, could endanger the passenger's safety and privacy, as well as result in tangible losses for data providers. In the finance domain, FL can be used for loan risk assessment (Cheng Y. Y., 2020). For example, WeBank, China's first Internet-only bank, uses a federated risk control system for small and micro enterprise loans. To further exemplify the role of FL in the energy space, a review of smart grid applications that have adopted Federated Learning as part of their solution strategy is presented in Table 13.

Table 13: Applications of Federated Learning

Citation	Keywords	Overview
(Su, 2021)	Smart Grid, Artificial Intelligence of Things (AIoT), Edge-Cloud Collaboration	The researchers proposed a FL-enabled AIoT scheme to preserve energy data privacy in edge-cloud collaboration environments. A two-layer, deep reinforcement-learning-based incentive algorithm was developed to encourage energy data owners (EDOs) to share high-quality data models. The scheme was simulated in an environment with an aggregator, four energy service providers (ESPs), and 50 individual EDOs. The simulation results showed that the scheme could improve the ESPs profit and reduce task latencies, justifying the value of sharing higher-quality models,
(Badr, 2023)	Energy Forecasting, Smart Grid	The researchers developed a federated learning-based energy prediction system aimed at achieving high accuracy while preserving the customer's private information. Inner-product functional encryption (IPFE) was used to parametrize the customer's model so that the data could be sent to the utility provider anonymously. The proposed scheme enabled the utility provider to use the encrypted parameters to build a global model.
(Saputra, 2019)	Electric Vehicle, Energy Demand, Charging Station, Clustering	A federated energy demand learning (FEDL) approach was used by researchers to reduce the communication costs between a charging station (CS) and its energy provider while preserving the privacy of the electric vehicle owner using the CS. A clustering-based EDL method was implemented using real data obtained from charging stations in Dundee city, United Kingdom. The dataset consisted of 65,601 transactions and 58 CSs. The results showed

		that the proposed approach increased energy demand by 24.63% and reduced communication overhead by 83.4% (when compared to other baseline machine learning algorithms).
(Wang X. X., 2021)	Electric Vehicle, Charging Station Recommendation System, Homomorphic Encryption	The researchers proposed an EV charge point recommendation framework based on FL that optimizes infrastructure usage while preserving the privacy of the underlying infrastructure and end-user user data (including EV-specific). A vertically-federated factorization machine algorithm based on HE was used to build a model that securely aggregates different data sources (charge points, users, and EVs). The results found that their model improved the area under the curve[1] metric by 6% over a traditional regression model while maintaining user privacy.
(Zhang L. J., 2023)	Photovoltaic Power Prediction, Concept Drift, Broad Learning System	The researchers developed a FL-based incremental broad learning system (BLS) and used it to address concept drift in photovoltaic (PV) power prediction. Concept drift is an ML term used to describe the loss of accuracy as the system evolves over time. The method was tested on a public dataset from the Desert Knowledge Australia Solar Centre. The results of the experiments found that their method greatly improved the prediction accuracy of the model and required less training time when compared to state-of-the-art learning algorithms while preserving data privacy.
(Liu Y. J., 2020)	Traffic Flow Prediction (TFP), Deep Learning, Gated Recurrent Unit (GRU)	A FL-based Gated Recurrent Unit (FedGRU) algorithm was proposed by the researchers to provide accurate forecasting of traffic flows while preserving commuter's privacy (e.g., license plates, vehicle location, etc.). Model parameters were transmitted using a federated averaging algorithm to lower transmission overheads, and a joint announcement protocol was implemented to improve scalability. The algorithm was tested on the Caltrans Performance Measurement System, a real-world data system with 39,000 sensors. The study determined that FedGRU reduced communication overheads by 64.10% while introducing a mean deviation of (0.76 km/h) when compared to state-of-the-art centralized methods, while providing data privacy.
(Li J. Y., 2021)	Smart Healthcare System, Alzheimer's disease detection	The researchers developed an Alzheimer's disease (AD) detection system (ADDetector) that implemented FL and DP to preserve the privacy of users' raw data and model details during data transmission. An asynchronous privacy-preserving aggregation framework was developed to secure the model aggregation process between the cloud server and client. The proposed system was evaluated with the ADReSS Challenge dataset. The results of the experiments showed that the proposed system maintained more than 78% accuracy in real-world scenarios and the time overhead was 0.7 seconds.

5.2 Challenges Faced by Federated Learning

While the user's raw data is kept under their control and not in a central server, FL cannot be considered secure or private by itself because an adversary may still infer information from

[1] The AUC is a measure of the true positive rate (sensitivity) against the false positive rate (1-specificity) for random samples. Higher values indicate better model accuracy.

the final model parameters if data dependencies remain[1]. Therefore, other privacy-preserving mechanisms such as secure multi-party computation, differential privacy, and homomorphic encryption must be leveraged in order to provide the necessary security and privacy guarantees. This may lead to a series of advantages and drawbacks that must be considered under the application-specific context. For example, integrating MPC into a FL scheme can reduce the efficiency of computations (e.g., not all algorithms benefit from distributed computations) or introduce communication overheads. Introducing DP into FL scheme requires application designers to determine the ideal trade-off between privacy vs accuracy, which ultimately impacts the model accuracy. When using HE methods, computational efficiency plays a key role, but it also requires establishing key management mechanisms (e.g. do all edge clients need a unique key pair to share data with the model creator, or can some clients share keys?).

In addition to the underlying privacy engine considerations, the successful management of communication costs and overheads, especially in network constrained environments, is one of the main challenges that FL faces in the communication domain (Banabilah, 2022). This is particularly true for low-bandwidth systems that contain a large number of participants, which require an equally large number of communication paths that could lead to network congestion issues. This might be particularly important for current-generation smart meter networks that operate at a fraction of the speed and bandwidth of a typical broadband Internet connection.

The reliability of end user's processing systems is another challenge for FL due to the heterogeneity of their hardware and network dependability (Banabilah, 2022). For example, some devices on the network will compute and transmit their updated model parameters at a slower rate than others, while others may drop from the network randomly. This requires FL solutions to adopt Crash Fault Tolerance (CFT) architectures that enable them to compute the end result while tolerating a certain ratio of non-respondent systems, or failed network links. Although CFT may also be used to ignore slow-responding systems and hence speed up processing, such optimization could inadvertently exclude slower participants, which may lead to underrepresentation and exclusions that disproportionality affect less capable systems.

A further, often overlooked issue is maintaining trust in a distributed environment. Although mechanisms such as public key cryptography can be used to authenticate participants, model updates (e.g., the actual model weights) by design must remain private, potentially leading to data poisoning attacks which may reduce the accuracy of the global model (Tolpegin, 2020). Such attacks could be mitigated by implementing outlier detectors, maintaining track of participating's historical reputation, or using peer-level reviews to reduce the chances of adopting incorrect model updates.

5.3 Computational Overheads of Federated Learning approaches

There are two challenges faced by FL when attempting to determine the computational overhead of various algorithms (Almanifi, 2023). The first challenge is that evaluations rely on a small pool of common datasets (e.g., MNIST, CIFAR10) that can skew the results to show that a FL model has higher prediction accuracy than it actually has for the target use case. This also makes it difficult to assess the effects of performance optimizations on the model's accuracy. Perhaps, the main reason is that it is difficult to recreate the "imperfect" nature of real-world

[1] Consider this example. If the global average of a group of 9 students is 3.4 and a new student updates the global average to 3.3, then the student must have had a GPA of 2.4 $(9 * 3.4 + 2.4)/10 = 3.3$, clearly bypassing any privacy expectations. Further examples are available in (Banabilah 2022)

systems, where participants may fail to work collaboratively (either due to random failures or lack of engagement). The second challenge is that there is not a standard set of metrics to benchmark FLs, other than accuracy metrics, communication characteristics or privacy attributes (which are equally hard to compare), which makes it difficult to make fair comparisons. The lack of standardization in benchmarking may be due to the heterogeneity of the data being modeled, the ML model characteristics, but also due to the devices that perform the training and updates. For example, in a smart grid network, the hardware of the smart meter may have vastly different capabilities than that of a consumer-grade computer. However, smart meters may be highly homogeneous within a service region (e.g., the same hardware vendor) making it possible for a utility to optimize a given solution but difficult for another utility to adopt the same solution (unless the same vendor is involved).

Based on this limitation, this report limits itself to discussing typical metrics that can be used to gauge a FL method's potential. Naturally, these metrics may need to be adapted to the unique characteristics of the deployment scenario.

- Time overhead metrics which includes 1) the time (t_{comp}) required by a client to perform a computation and 2) the time (t_{comm}) taken by the client to communicate with the central server (to download/upload model parameters) (Luo, 2021).Typical training time (per round) that is composed of t_{comp} times the number of local training interations summed with t_{comm}.

- Energy cost (e_{comp}) of a client to complete a computation and the energy cost (e_{comm}) to communicate with the central server.

- Typical energy cost (per round) that is composed of e_{comp} times the number of local training iterations summed with t_{comm}.

- Memory consumption of a client completing a computation for a pre-selected number of rounds (Gao, 2020).

- Accuracy of the trained global model (Zhang Z. Z., 2020).

- Convergence speed and efficiency.

6.0 Conclusion and recommendations

This report presented a review of Privacy Enhancing Techniques in the context of the smart grid domain. Each of these techniques was contrasted in terms of their key capabilities, typical uses, and known limitations. Such descriptions are intended to enable application engineers to identify the technique (or techniques) that are aligned with their use-case needs. Furthermore, this report discussed the time and space complexities associated with the adoption of such techniques, a discussion that is relevant due to the limited computational and communication capabilities that are available in smart grid deployments. The techniques can be summarized as follows:

- Homomorphic encryption uses encryption to enable computations to be performed on previously encrypted data without requiring decryption, which lowers its exposure by reducing the number of situations in which the data can be viewed in unencrypted form. However, this security feature significantly increases the computational overhead when compared to other mechanisms.

- Zero-knowledge proofs (ZKP) are specifically designed to enable a user to prove to an external entity that they know a secret without leaking any data other than the fact that they know the secret. They offer a relatively low overhead cost when compared to other public key cryptographic methods but a limited set of functions that can be implemented.

- Differential privacy is a formal mathematical construct that guarantees that the inclusion or exclusion of a single data point cannot be determined from the aggregated dataset. By preventing inferences at the individual level, it grants strong privacy protections. However these protections are based on the introduction of noise, which can lower data utility, in addition complete privacy can only be guaranteed under certain conditions.

- Federated learning is a method of machine learning that enables increased privacy of user data by not requiring users to upload their raw data to a central server for training. Instead, the process of training the global model consists of incremental updates that are exchanged between the central server and remote users to assemble a global model via collaboration. The main drawback is that it cannot be considered secure by itself, thus requiring other privacy-preserving mechanisms to accomplish data privacy.

Similarly, the main contributions of this work can be summarized as follows:

- Advocates for the adoption of privacy-preserving mechanisms as an essential tool for achieving a holistic approach to data privacy in the smart grid (but also applicable to the energy sector in general).

- Provides researchers, engineers, and system designers with a practical overview of PET by summarizing strengths, typical use cases, and known limitations.

- Presents a comprehensive survey of the computational burdens associated with implementing PET technologies under the context of the typical smart grid deployment.

- Identifies technological limitations that currently act as barriers towards the larger scale adoption of these technologies.

Based on the current state of the art, this report further recommends:

- **Fostering research and development of privacy-preserving technologies:** Support the advancement of privacy-preserving technologies that are specifically tailored to address the engineering challenges of the smart grid. This may involve addressing the communication and computational barriers, streamlining solution integration, and developing standardized benchmarking methods. When combined, these efforts will help to facilitate the widespread adoption of these technologies.
- **Encouraging the adoption of privacy-by-design principles**: Promote the integration of privacy-by-design principles throughout an application's development and operational lifecycle. This can be done by communicating both the technical and societal benefits to stakeholders, ensuring the value proposition of privacy stands on its own. The aim is to foster an ecosystem where user data is not only valuable from an engineering perspective but also meets end-user expectations for privacy and security.
- **Advocating for policy development and regulatory compliance**: Promote the development of new policies and encourage the adoption of regulatory standards within the energy sector. The industry can benefit from adapting existing regulatory frameworks, such as the Health Insurance Portability and Accountability Act (HIPAA) from the healthcare domain, to expedite maturity while minimizing potential missteps. Utilities must stay informed about legislative changes and integrate these technologies to avoid legal repercussions and enhance public trust.

Adopting a privacy-aware smart grid will necessitate advances in both policy and technical domains. From a technical perspective, it is crucial for researchers to address the inherent challenges associated with the practical adoption of privacy-enhancing technologies (PETs). This effort needs to be complemented by educational initiatives that raise awareness about the advantages and disadvantages of PETs, and how these may impact the end application. Furthermore, security considerations must be thoroughly addressed before implementing PET algorithms to ensure the resulting applications continue to provide sufficient security guarantees over the long term.

To overcome these challenges, stakeholders need to be well-informed of the capabilities, drawbacks, and benefits of PETs. Promoting collaboration among stakeholders can ensure that applications are designed and implemented with potential weaknesses in mind. Effective approaches will likely start by identifying and prioritizing problems and then finding suitable mechanisms to address them. For example, Table 14 provides a list of challenges within the smart grid, potential solutions (e.g., the tools that are available), and the limitations or risks that may be introduced by adopting a particular solution. Although, the examples are not intended to be authoritative, such a list can be further refined by classifying issues according to the stakeholders involved or by distinguishing between the different spatio-temporal characteristics of the problem.

Table 14: An overview of challenges, solutions, and their associated drawbacks in the field of energy domain.

DSO model	Problem	Solutions	Issues
Trusted	Enabling consumers to audit if data usage aligns with prior consent	Digital/physical consent forms	No automatic guarantee on data usage alignment, violations may be legally enforceable.
		Third party verifications	Providing data access to third parties either increases or shifts risk.
Trusted	Ensuring data integrity (in centralized or high trust environments)	PKI-based digital signing approaches	PKIs are tied to real-world identities, a mapping that may be abused.
		Zero Knowledge Proofs	ZKP computation cost may be prohibitive in resource constrained environments.
Both	Ensuring data integrity (in decentralized or low trust environments)	Anonymous authentication or identity stripping	Compliance may be hard to track, actors may become transient. Anonymization does not guarantee privacy.
		Permissioned DLT	Must develop trust metrics (e.g., based on past performance). Scalability issues (network size and commit throughput)
		Permissionless DLT	Most implementations are energy inefficient, prone to sybil attacks (a fake majority overrides consensus)
Trusted	Auditing participants (in centralized or high trust environments)	Digital signed events (e.g. via PKI certificates)	PKIs are tied to physical identities, requires periodic signing of messages (e.g. every hour)
		Authentication-based systems	All data must be transmitted. Although data may be encrypted, attacks to these types of systems are common.
Both	Auditing (in decentralized or low trust environments)	Homomorphic Encryption	May come with computational or algorithmic limitations that limit its use.
		Zero Knowledge Proofs (ZKP)	The types of audits possible are tied to ZKP-specific capabilities. Interactive ZKPs may be abused.
		Differential privacy	May introduce noise that exceeds application's tolerance.
		Threshold cryptography	Probabilistic protection mechanism. Implementing a distributed key generation mechanism is hard.
Both	Billing (with load hiding mechanism)	Virtual battery models or physical storage	May hide peaks or valleys in the demand curve. Once (real or virtual) capacity limits are reached, no additional protection is possible.
		Delayed aggregation	May reduce the ability for applications to react in real time.
Untrusted	Billing (via secure computation/privacy methods)	Zero-Knowledge Proofs	Computationally intensive. Interactive approaches may be abused, negating the method's utility
		Homomorphic encryption	The number of supported operations is low (algorithm dependent), may limit data repurposing. Significant computational overheads
		Differential privacy	Noise added may exceed application tolerance, lowering data utility.
Trusted	Billing (via secure computation/privacy methods)	Digital signed events (e.g., via PKI certificates)	PKIs are tied to physical identities. Message contents may be encrypted but do not offer formal privacy guarantees.
		Authentication-based systems	All data must be transmitted; although data may be encrypted, data exfiltration and key theft remain risks.
Both	Enabling resource	Homomorphic encryption	Computationally intensive, hard to share results with multiple entities (secure key distribution challenge).

	coordination (in decentralized or low trust environments)	Multi-party computation	Must be built upon privacy or security constructs to achieve goal. Scalability is algorithm-dependent.
		Differential privacy	High noise when the number of aggregated actors or values is low or when DP must be applied locally. Noise characteristics can be improved via trusted aggregator
		Permissionless DLT	Vulnerable to sybil attacks (i.e. ill clusters manipulate consensus). Consensus algorithm defines efficiency. Identity anonymization does not equal privacy.
		Permissioned DLT (with ledger segmentation)	Identities are known within a segment. Ledger data is visible to segment's participants. Inter-ledger or inter-segment coordination may introduce centralization.
Both	Real time monitoring and or supervision	Centralized aggregation with cybersecurity	Insider threats may lead to data exposure. No guarantees on the use of data or availability to third parties. If leaked, raw data may become publicly accessible.
		Cryptography based PETs	Computational efficiency may limit responsiveness and scalability. Some mechanisms require key distribution.
		Differential privacy	Trusted aggregators must securely discard raw data and correctly apply privacy mechanism. If DP is applied locally, greater amounts of noise will be introduced.
Both	Building ML solutions (in decentralized or low trust environments)	Federated learning	Must adopt a privacy construct (in the input and output phases) to mitigate against training data extraction. Addressing model poisoning is hard. Scalability remains an issue due to communication and computation overheads.
Both	Ensuring long term data storage	Traditional cryptographic methods	Key management is an issue. Once a key is compromised, protections afforded by encryption using that key are useless (assuming encrypted data is also accessible).
		Cryptography based PETs	Key management and zero-day vulnerabilities may be an issue. Computational efficiency may become a problem of scale.
		Differential privacy	Long-term data series or repetitive querying may eventually exceed the allotted privacy budget. Truly independent datasets are hard to find, hidden dependencies may lower DP effectiveness.

Appendix A – Digital watermarking

Digital watermarking is the process of embedding a hidden signal that is hard to detect or modify without knowing the underlying security mechanism. Most modern approaches leverage bit-level modifications that do not degrade or affect the original media intent (e.g., digitally encoded images, audio, and video signals) while being able to embed information defined by the content creator (Singh P. a., 2013). Watermarks are often used for copyright protection and authentication in the media industry, and hence they appeared as a promising solution to facilitate data tracking and ensure data use was in accordance with the end-user intended consent. Once an in depth-review was commenced, it became apparent that most research on "privacy" and "digital watermarks" were with regards to preserving the privacy of the watermark itself rather than the data, and many of the applications were related to copyright protection or to ensure tamper resistance, with few actual privacy applications. A few examples of articles found related to providing access controls were (Kountchev, 2015), (Yang H. a., 2015), and (Guo J. W., 2018). In (Kountchev, 2015), multiple fragile watermarks were used as decomposition layers in an inverse pyramid decomposition scheme and were applied to medical images. Deeper layers in the pyramid provided higher quality images to a user that had the required permissions to access it. In (Yang H. a., 2015), the researchers presented a method of applying a binary watermark sequence into a visibly watermarked block truncation coding (BTC) compressed image to prevent unauthorized users from recovering the original pixels in the watermarked region. In (Guo J. W., 2018), a model that embedded access control policies as watermarks onto an image was proposed to solve the issue of the image's original access controls being removed when being redistributed by a server.

Appendix B – Examples

B.1 Asymptotic Notation

Asymptotic notation (aka O notation) is a fundamental concept in computer science used to describe the complexity of an algorithm (and thus its efficiency). By using a simple mathematical expression, asymptotic notation it is able to capture an algorithm's performance in terms of how its runtime or space requirements grow as the input size increases (which is usually denoted by n). By abstracting away constants and lower-order terms, asymptotic notation allows for a clear comparison of the scalability and efficiency of different algorithms.

During this report, many tables made use of asymptotic notation to describe an algorithm's expected performance under the average, worst, or ideal conditions. These are commonly known by the names Big Theta (Θ), Big-O (O), and Big Omega (Ω) respectively. A high-level overview of their definition and how they are determined is discussed in the remainer of this section.

B.1.1 Big O notation

Big-O (O) notation describes the tight upper bound of an algorithm's runtime and is defined as:

"Let $f(n)$ and $g(n)$ be functions that map positive integers to positive real numbers. We say that $f(n)$ is $O(g(n))$ if there exists a real constant $c > 0$ and there exists an integer constant $n_0 \geq 1$ such that $f(n) \leq c * g(n)$ for every integer $n \geq n_0$" (McCann, 2009).

(McCann, 2009) provides the following example of determining whether the function $f(n) = 7n + 8$ is $O(g(n))$, where $g(n) = n$:

$$7n + 8 \leq cn \text{ ; } let\ c = 8$$

$$7n + 8 \leq 8n$$

$$7n - 8n + 8 \leq 0$$

$$-n \leq -8$$

$$n \geq 8$$

Thus, there exists a constant $c = 8$ and a constant $n_0 = 8$ such that $f(n)$ is $O(g(n)) = O(n)$ for all integers greater than or equal to 8. Sometimes, the big-O runtime of an algorithm can be determined without mathematical proofs. For example, suppose we have an array of *n* items consisting of unique integers and we want to determine if 7 is in the array. An algorithm has the worst runtime when 7 is either not in the array or if it is the last item inspected. To describe this aspect of the algorithm, we would say it runs in $O(n)$.

B.1.2 Big-Omega

To describe the best-case runtime of an algorithm, big-omega (Ω) notation is used and is formally defined as:

"Let $f(n)$ and $g(n)$ be functions that map positive integers to positive real numbers. We say that $f(n)$ is $\Omega(g(n))$ if there exists a real constant $c > 0$ and there exists an integer constant $n_0 \geq 1$ such that $f(n) \geq c \cdot g(n)$ for every integer $n \geq n_0$" (McCann, 2009).

For example, let $f(n) = 4n$ and let $g(n) = n$, is $f(n)$ equal to $\Omega(n)$?

$$4n \geq cn \text{ ; } let\ c = 3$$

$$4n \geq 3n$$

We can immediately see that with a constant value $c = 3$, no matter what value we select for *n*, $f(n) = 4n$ will always be greater than or equal to $3n$. Thus, $f(n) = 4n$ can be described as having $\Omega(n)$ runtime. Like big-O, sometimes we can determine big-omega without mathematical proofs. Returning to the example of searching an array of n items for the integer "7", the best possible case for the algorithm is when "7" is the first item inspected and found immediately in constant time. This gives the algorithm a big-omega time complexity of $\Omega(1)$. Big-omega is generally not used to describe algorithms but is used to define the notation of big-theta (Θ) (McCann, 2009).

B.1.3 Big-theta

Big-theta describes the situation where the function $g(n)$ is both the tight upper and lower bounds of an algorithm. It is defined as follows:

"Let $f(n)$ and $g(n)$ be functions that map positive integers to positive real numbers. We say that $f(n)$ is $\Theta(g(n))$ if and only if $f(n) \in O(g(n))$ and $f(n) \in \Omega(g(n))$" (McCann, 2009).

For example, if $f(n) = 4n$ and $g(n) = n$, then we know from the previous example that $f(n)$ is $\Omega(n)$. In determining big-O time complexity of $f(n) = 4n$, if the constant value c was 5 instead of 3, we'd have the equation: $4n \leq 5n$. Thus, for all values of n when $c = 5$, $f(n)$ is less than or equal to $5n$. Thus, $f(n) = 4n$ has $O(n)$ time complexity as well. Since $f(n) = O(n)$ and $f(n) = \Omega(n), f(n) = \Theta(n)$.

B.2 Sample HE

Using the base example presented in (Mallouli, 2019), which describes the traditional RSA algorithm, it is possible to construct a multiplicative HE example.

The first step in generating the key pair is selecting two prime numbers p and q.

$$p = 5, q = 7$$

Then a modulus used in both the public and private keys is generated through the product pq.

$$n = pq$$

$$n = 5 * 7;\ n = 35$$

A value m is calculated as $(p - 1)(q - 1)$ and a value e is chosen such that it is not a factor of m.

$$m = (p-1)(q-1)$$

$$m = 4 * 6;\ m = 24$$

$$e = 5$$

A private key d is then calculated as $(de)\ mod\ m\ =\ 1$.

$$de\ mod\ m = 1$$

$$d(5)\ mod\ 24 = 1$$

$$d = 29$$

The public key $\{n, e\}$ is $\{35, 5\}$ and the private key $\{n, d\}$ is $\{35, 29\}$ in this example. The ciphertext c is generated through the computation of the plaintext M raised to the power of e mod n. Given two plaintexts $M_1\ =\ 5$ and $M_2 = 6$:

$$c_1 = M_1^e\ mod\ n$$

$$c_1 = 5^5\ mod\ 35$$

$$c_1 = 3125 mod\ 35;\ c_1 = 10$$

And similarly:

$$c_2 = M_2^e\ mod\ n$$

$$c_2 = 6^5\ mod\ 35$$

$$c_2 = 7776\ mod\ 35;\ c_2 = 6$$

For simplicity, lets assume that an untrusted agent is tasked with multiplying c_1 and c_2, since both messages are encrypted then the only thing that the external agent can learn is that $c_3 = c_1 * c_2$ is 60, but it cannot discover the unencrypted value.

The decryption of c_3 is obtained through the computation of the ciphertext c_3 raised to the power of $d\ mod\ n$. Give a ciphertext $c\ =\ 60$:

$$M = c^d\ mod\ n$$

$$M = 60^{29}\ mod\ 35; M = 60 * 60^{28}\ mod\ 35$$

$$M = 60 * (60^4)^7\ mod\ 35$$

$$M = 60 * (12960000)^7\ mod\ 35$$

$$M = 60 * (12960000\ mod\ 35)^{mod\ 35}$$

$$M = 60 * 25^7 \; mod \; 35$$

$$M = 60 * 6{,}103{,}515{,}625 \; mod \; 35$$

$$M = 366210937500 \; mod \; 35$$

$$M = 30$$

B.3 Sample ZKP

A highly simplified example of the inner workings of ZKP is presented in (NIST, 1991) and has been reproduced here for simplicity:
The authority decides on a number N used for everyone, e.g., take $N = 77$ $(7x11)$. Everyone knows this number. The authority may then choose two numbers which form an ID for Alice. Suppose these are $\{58,67\}$. Everyone knows Alice's ID. The authority then computes two other numbers $\{9,10\}$ which are given to Alice alone; she keeps these private. The latter numbers were chosen because $9^2 * 58 = 1(mod\ 77)$ and $10^2 * 67\ 1\ (mod\ 77)$.
Now Alice can identify herself to Bob by proving that she possesses the secret numbers {9,10} without revealing them. Each time she wishes to do this she can choose some random numbers such as {19,24,51} and compute:

$$19^2 = 53 \ (mod\ 77),$$
$$24^2 = 37 \ (mod\ 77),$$
$$51^2 = 60 \ (mod\ 77).$$

Alice then sends {53,37,60} to Bob. Bob chooses a random 3 by 2 matrix of 0's and 1's, e.g.,

$$E = \begin{matrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{matrix}$$

Bob sends E to Alice. On receipt, Alice computes

$$19 * 9^0 * 10^1 = 36 \ (mod\ 77),$$
$$24 * 9^1 * 10^0 = 62 \ (mod\ 77),$$
$$51 * 9^1 * 10^1 = 47 \ (mod\ 77).$$

Alice sends $\{36,62,47\}$ to Bob. Finally, Bob can check to see that Alice is who she says she is. He does this by checking that:

$$36^2 * 58^0 * 67^1 = 53 \ (mod\ 77),$$
$$62^2 * 58^1 * 67^0 = 37 \ (mod\ 77),$$
$$47^2 * 58^1 * 67^1 = 60 \ (mod\ 77).$$

The original numbers $\{53,37,60\}$ that Alice sent reappear. Actually, this doesn't really prove Alice's identity; she could have been an impersonator. But the chances of an impersonator succeeding would have only been 1 in 64.

B.4 Sample DP

(Jiang H. J., 2020) provides a simple example of the random response perturbation mechanism for achieving local DP. Given n users with an unknown proportion π that are diseased, a survey is issued to n users to determine π by asking if they have a disease, to which a user may answer “yes” or “no”. A user can flip a coin that has a probability p of landing heads up and a probability $1 - p$ of it landing tails up to determine whether they answer truthfully. If the coin shows heads, the user will answer truthfully, and they will lie if it shows tails. The level of local DP provided by this technique is:

$\epsilon = \left|\ln\left(\frac{p}{1-p}\right)\right|$; Where ϵ is the privacy budget.

An example of a randomized response process can be observed in Figure B.1, where a subject starts by flipping a coin. If the coin lands on heads, they answer the question truthfully. If the coin lands on tails, then the subject must flip the coin again and use the result to answer the question. Such a process offers the subject the ability to deny any recorded answer (and instead assert it was a random response), while still producing meaningful results at the population level (e.g., the effects of the coin flip can be cancelled out).

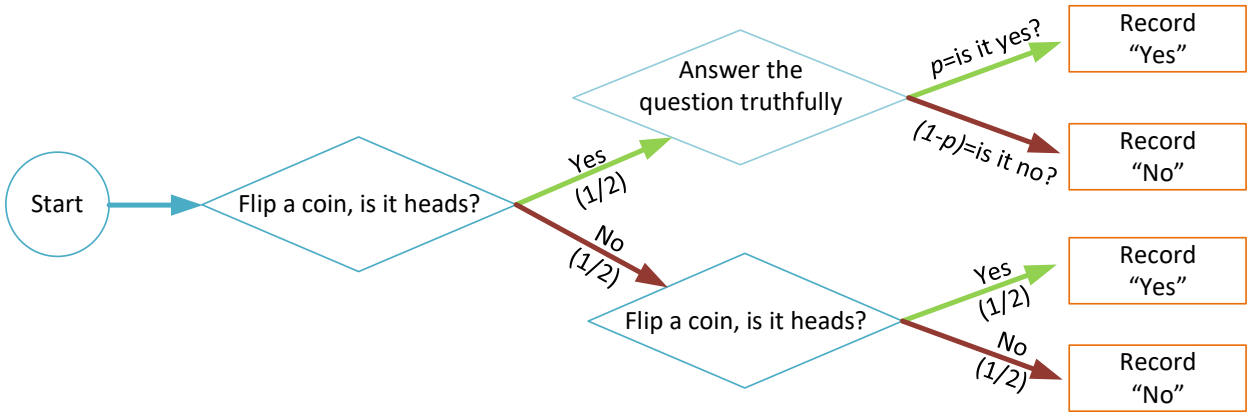


Figure B.1. The probability tree in a randomized response survey, assuming a fair coin is used, from (Sebastian Cardenas, Mukherjee, & Ramirez, 2023).

Based on the probability tree presented in Figure B.1, it can be shown that the expected number of true "Yes" answers can be modeled by $Pr(yes) = \left(\frac{1}{4}\right)(1-p) + \left(\frac{3}{4}\right)p = \left(\frac{1}{4}\right) + \frac{p}{2}$. Therefore, for a significantly large population, P can be estimated as $2\left(\frac{\#yes}{pop.\ size} - ¼\right)$.

B.5 Sample FL

An example of the Federated Averaging (FedAvg) algorithm is described in (Nilsson, 2018) as follows: A central server contains a shared global model w_t, where t is the current iteration of the communication round. The algorithm consists of five hyperparameters that control the learning process: the fraction of clients C that will be chosen to train the model locally, the local mini-batch size B, the number of epochs E that the clients will spend training the local model, a learning rate η, and a learning rate decay of λ. Optimization of the algorithm is performed on the client side via Stochastic Gradient Decent (SGD). The algorithm first initializes the global model w_0 and then each round of communication consists of the following steps:

1. A subset of clients S_t are selected such that $|S_t| = CK \geq 1$, where K is the total number of clients in the network. The server transmits the global model w_t to the selected clients S_t.

2. Each client trains their local model w_t^k on their raw data and updates the shared model, $w_t^k \leftarrow w_t$, by grouping their local data into batches of size B and applying E epochs of SGD.

3. The client uploads their trained model w_{t+1}^{k} to the server, which generates a new global model w_{t+1} by calculating the weighted sum of all the selected client's models through the equation:

$$w_{t+1} = \sum k \in S_t \frac{\mathrm{n_k}}{\mathrm{n_\sigma}} w_{t+1}^{k}$$

Where $\mathrm{n_k}$ is the number of data points on client K and $\mathrm{n_\sigma} = \sum k \in S_t \mathrm{n_k}$.

7.0 References

Abdalzaher, M. S. (2022). Data privacy preservation and security in smart metering systems. *Energies 15, no. 19*, 7419.

Acar, A. H. (2018). A survey on homomorphic encryption schemes: Theory and implementation. *ACM Computing Surveys (Csur) 51, no. 4*, 1-35.

Alamatsaz, N. A. (2017). Towards Efficient Privacy-Preserving Data Aggregation for Advanced Metering Infrastructure. *International Journal of Computer Networks & Communications Vol.9, No. 5*.

Alharbi, A. H. (2020). Survey on homomorphic encryption and address of new trend. *International Journal of Advanced Computer Science and Applications 11, no. 7*.

Alisic, R. M. (2020). Bounding privacy leakage in smart buildings. *arXiv preprint arXiv:2003.13187*.

Almadhoun, N. E. (2020). Differential privacy under dependent tuples—the case of genomic privacy. *Bioinformatics 36, no. 6*, pp. 1696-1703.

Almanifi, O. R.-O.-L. (2023). Communication and computation efficiency in federated learning: A survey. *Internet of Things 22*, 100742.

Asghar, M. R. (2017). Smart meter data privacy: A survey. *IEEE Communications Surveys & Tutorials 19, no. 4*, 2820-2835.

Awadallah, R. a. (2020). Homomorphic encryption for cloud computing and its challenges. *In 2020 IEEE 7th International Conference on Engineering Technologies and Applied Sciences (ICETAS)* (pp. 1-6). Kuala Lumpur: IEEE.

Badr, M. M. (2023). Privacy-preserving and communication-efficient energy prediction scheme based on federated learning for smart grids. *IEEE Internet of Things Journal*.

Baghery, K. M. (2021). Another look at extraction and randomization of Groth's zk-SNARK. *In Financial Cryptography and Data Security: 25th International Conference* (pp. 457-475). Springer Berlin Heidelberg.

Banabilah, S. M. (2022). Federated learning review: Fundamentals, enabling technologies, and future applications. *Information processing & management 59, no. 6*, 103061.

Bassily, R. a. (2015). Local, private, efficient protocols for succinct histograms. *In Proceedings of the forty-seventh annual ACM symposium on Theory of computing* (pp. 127-135). Portland: Association for Computing Machinery.

Boneh, D. E.-J. (2005). Evaluating 2-DNF formulas on ciphertexts. *Theory of Cryptography: Second Theory of Cryptography Conference, TCC 2005* (pp. 325-341). Cambridge: Springer Berlin Heidelberg.

Boudot, F. (2000). Efficient proofs that a committed number lies in an interval. *In International Conference on the Theory and Applications of Cryptographic Techniques* (pp. 431-444). Berlin: Springer Berlin Heidelberg.

Chamikara, M. A. (2020). Privacy preserving face recognition utilizing differential privacy. *Computers & Security 97*, 101951.

Chen, Z. Y. (2023). A survey on zero-knowledge authentication for internet of things. *Electronics 12, no. 5*, 1145.

Chenal, M. a. (2015). On key recovery attacks against existing somewhat homomorphic encryption schemes. *In Progress in Cryptology-LATINCRYPT 2014: Third International Conference on Cryptology and Information Security in Latin America Florianópolis, Brazil, September 17–19, 2014 Revised Selected Papers 3* (pp. 239-258). Brazil: Springer International Publishing.

Cheng, Y. Y. (2020). Federated learning for privacy-preserving AI. *Communications of the ACM 63, no. 12*, 33-36.

Cheng, Z. F.-Y. (2021). A homomorphic encryption-based private collaborative distributed energy management system. *IEEE Transactions on Smart Grid 12, no. 6`*, 5233-5243.
Cormode, G. T. (2018). Marginal release under local differential privacy. *In Proceedings of the 2018 International Conference on Management of Data* (pp. 131-146). New York: Assocation for Computing Machinery.
Craig, G. (2009). A fully homomorphic encryption scheme. *Diss. Stanford University*.
Culnan, M. J., & Bies, R. J. (2003). Consumer Privacy: Balancing Economic and Justice Considerations. *Journal of Social Issues*, 323-342.
Currie, R. S. (2023). Data privacy for the grid: Toward a data privacy standard for inverter-based and distributed energy resources. *IEEE Power and Energy Magazine 21, no. 5*, 48-57.
Deng, C. J. (2019). A survey on range proof and its applications on blockchain. *In 2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)* (pp. 1-8). Guilin: IEEE.
Diffie, W. a. (1976). New directions in cryptography. *IEEE Transactions of Information Theory 22, 6*, pp. 644-654.
Dileep, G. J. (2020). A survey on smart grid technologies and applications. *Renewable energy 146*, 2589-2625.
Doan, T. V.-L. (2023). A survey on implementations of homomorphic encryption schemes. *The Journal of Supercomputing 79, no. 13*, 15098-15139.
Dwork, C. (2006). Differential privacy. *In International colloquium on automata, languages, and programming* (pp. 1-12). Berlin: Springer Berlin Heidelberg.
Dwork, C., & McSherry, F. (2011). Differential Privacy – A Primer for the Perplexed. *CONFERENCE OF EUROPEAN STATISTICIANS .* UNITED NATIONS ECONOMIC COMMISSION.
Dwork, C., Kohli, N., & Mulligan, D. (2019). Differential privacy in practice: Expose your epsilons! *Journal of Privacy and Confidentiality*.
Eibl, G. K.-W. (2018). The influence of differential privacy on short term electric load forecasting. *Energy Informatics 1, no. Suppl 1*, 48.
ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory 31, no. 4*, 469-472.
Erlingsson, Ú. V. (2014). Rappor: Randomized aggregatable privacy-preserving ordinal response. *In Proceedings of the 2014 ACM SIGSAC conference on computer and communications security* (pp. 1054-1067). Scottsdale: Association for Computing Machinery.
Fan, J. Q. (2017). Privacy disclosure through smart meters: Reactive power based attack and defense. *In 2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)* (pp. 13-24). Denver: IEEE.
Fanti, G. V. (2015). Building a RAPPOR with the unknown: Privacy-preserving learning of associations and data dictionaries. *arXiv preprint arXiv:1503.01214*.
Fellows, M. a. (1994). Combinatorial cryptosystems galore! *Contemporary Mathematics 168*, 51.
Finster, S. a. (2015). Privacy-aware smart metering: A survey. *IEEE communications surveys & tutorials 17, no. 2*, 1088-1101.
Gaba, G. S. (2022). Zero knowledge proofs based authenticated key agreement protocol for sustainable healthcare. *Sustainable Cities and Society 80*, 103766.
Gabay, D. K. (2019). A privacy framework for charging connected electric vehicles using blockchain and zero knowledge proofs. *In 2019 IEEE 44th LCN symposium on emerging topics in networking (LCN Symposium)* (pp. 66-73). Osnabrueck: IEEE.
Gai, N. K. (2022). An efficient data aggregation scheme with local differential privacy in smart grid. *Digital Communications and Networks 8, no. 3*, 333-342.
Ganesh, A. (2022). *Privacy, Improved Algorithms and Upper Bounds in Differential.* University of California, Berkeley.

Gao, Y. M. (2020). End-to-end evaluation of federated learning and split learning for internet of things. *arXiv preprint arXiv:2003.13376*.

Groth, J. (2009). Linear algebra with sub-linear zero-knowledge arguments. *In Annual International Cryptology Conference* (pp. 192-208). Berlin: Springer Berlin Heidelberg.

Guo, J. M. (2021). A practical privacy-preserving publishing mechanism based on personalized k-anonymity and temporal differential privacy for wearable IoT applications. *Symmetry 13, no. 6*, 1043.

Guo, J. W. (2018). A Watermark-Based In-Situ Access Control Model for Image Big Data. *Future Internet*, 69.

Haeberlen, A. B. (2011). Differential privacy under fire. *In 20th USENIX Security Symposium (USENIX Security 11).* San Francisco.

Hassan, M. U. (2019). Differential privacy for renewable energy resources based smart metering. *Journal of Parallel and Distributed Computing 131*, 69-80.

Ho, J. C.-Y. (2021). An anonymous on-street parking authentication scheme via zero-knowledge set membership proof. *arXiv preprint arXiv:2108.03629*.

Hossain, M. B. (2021). Cost-friendly differential privacy of smart meters using energy storage and harvesting devices. *IEEE Transactions on Services Computing 15, no. 5*, 2648-2657.

Husnoo, M. A. (2021). Differential Privacy for IoT-Enabled Critical Infrastructure: A Comprehensive Survey. *IEEE Access 9*, 153276-153304.

IEEE digital privacy group. (2023). *What Is Digital Privacy and Its Importance?*

Jiang, B. J. (2021). Differential privacy for industrial internet of things: Opportunities, applications, and challenges. *IEEE Internet of Things Journal 8, no. 13*, 10430-10451.

Jiang, H. J. (2020). Differential privacy and its applications in social network analysis: A survey. *arXiv preprint arXiv:2010.02973*.

Jiang, K. D.-L. (2013). Publishing trajectories with differential privacy guarantees. *In Proceedings of the 25th International conference on scientific and statistical database management*, (pp. 1-12).

Ju, Z. a. (2023). Local differential privacy-based privacy-preserving data range query scheme for electric vehicle charging. *IEEE Transactions on Network Science and Engineering*.

Kountchev, R. M. (2015). Content protection and hierarchical access control in image databases. *2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA)* (pp. 1-6). Madrid: IEEE.

Kumar, P. Y. (2019). Smart grid metering networks: A survey on security, privacy and open research issues. *IEEE Communications Surveys & Tutorials 21, no. 3*, 2886-2927.

Lee, J. S. (2019). From discovery to practice and survivorship: building a national real-world data learning healthcare framework for military and veteran cancer patients. *Clinical Pharmacology & Therapeutics 106, no. 1*, 52-57.

Lee, J.-W. H. (2022). Privacy-preserving machine learning with fully homomorphic encryption for deep neural network. *IEEE Access 10* , 30039-30054.

Lei, Y.-T. C.-Q.-S.-Q. (2022). A Renewable Energy Microgrids Trading Management Platform Based on Permissioned Blockchain. *Energy Economics*, 106375.

Leluc, R., Chedemail, E., Kouande, A., Nguyen, Q., & Andriamandratomanana, N. (2022). *Fully homomorphic encryption and bootstrapping.*

Li, J. Y. (2021). A federated learning based privacy-preserving smart healthcare system. *IEEE Transactions on Industrial Informatics 18, no. 3*.

Li, Q. Z. (2021). A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering 35, no. 4*, 3347-3366.

Liu, C. S. (2016). Dependence makes you vulnberable: Differential privacy under dependent tuples. *NDSS, vol. 16*, 21-24.

Liu, J. K.-K. (2020). Privacy-preserving COVID-19 contact tracing app: a zero-knowledge proof approach. *Cryptology ePrint Archive*.

Liu, S. G. (2023). An anonymous authentication mechanism based on zero-knowledge proof for power system. *In Second International Symposium on Computer Applications and Information Systems (ISCAIS 2023)* (pp. 28-38). Chengdu: SPIE.

Liu, Y. J. (2020). Privacy-preserving traffic flow prediction: A federated learning approach. *IEEE Internet of Things Journal 7, no. 8*, 7751-7763.

Lu, Y. X. (2020). ederated Learning for Data Privacy Preservation in Vehicular Cyber-Physical Systems. *IEEE Network 34, no. 3*, 50-56.

Luo, B. X. (2021). Cost-effective federated learning design. *In IEEE INFOCOM 2021-IEEE Conference on Computer Communications* (pp. 1-10). IEEE.

Mallouli, F. A. (2019). A survey on cryptography: comparative study between RSA vs ECC algorithms, and RSA vs El-Gamal algorithms. *In 2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)* (pp. 173-176). Paris: IEEE.

Marcolla, C. V. (2022). Survey on fully homomorphic encryption, theory, and applications. *Proceedings of the IEEE 110, no. 10*, 1572-1609.

McCann, L. (2009, February). *Asymptotic Notation: O(), o(), Ω(), ω(), and Θ().* Retrieved May 14, 2024, from The University of Arizona: https://www2.cs.arizona.edu/classes/cs345/summer14/files/bigO.pdf

McSherry, F., & Talwar, K. (2008). Mechanism Design via Differential Privacy. *48th Annual IEEE Symposium on Foundations of Computer Science .* Providence, RI, USA.

Miller, J. (2022, April 15). *The Frozen Heart vulnerability in Bulletproofs*. Retrieved from https://blog.trailofbits.com/2022/04/15/the-frozen-heart-vulnerability-in-bulletproofs/

Mohammed, N. S. (2015). Secure and private management of healthcare databases for data mining. *In 2015 IEEE 28th International Symposium on Computer-Based Medical Systems* (pp. 191-196). Sao Carlos: IEEE.

Morais, E. T. (2019). A survey on zero knowledge range proofs and applications. *SN Applied Sciences 1*, 1-17.

Mousavian, S. J. (2013). Real-time data reassurance in electrical power systems based on artificial neural networks. *Electric Power Systems Research 96*, 285-295.

Near, J. P., & Abuah, C. (2024). *The Exponential Mechanism*. Retrieved from https://programming-dp.com/ch9.html

Nilsson, A. S. (2018). A performance evaluation of federated learning algorithms. *In Proceedings of the second workshop on distributed infrastructures for deep learning* (pp. 1-8). Rennes: Association for Computing Machinery.

NIST. (1991). *NIST Special Publication 800-2: Public Key Cryptography (Archived).*

Paillier, P. (1999). Public-key cryptosystems based on composite degree residuosity classes. *In International conference on the theory and applications of cryptographic techniques* (pp. 223-238). Berlin: Springer Berlin Heidelberg.

Parker, K., Hale, M., & Barooah, P. (2022). Spectral Differential Privacy: Application to Smart Meter Data. *IEEE Internet of Things Journal*.

Partala, J. T. (2020). Non-interactive zero-knowledge for blockchain: A survey. *IEEE Access 8*, 227945-227961.

Podschwadt, R. D. (2022). A survey of deep learning architectures for privacy-preserving machine learning with fully homomorphic encryption. *IEEE Access 10*, 117477-117500.

Pop, C. D. (2020). Blockchain and demand response: Zero-knowledge proofs for energy transactions privacy. *Sensors 20, no. 19*, 5678.

Qiu, R. X. (2021). Differential privacy EV charging data release based on variable window. *PeerJ Computer Science 7*, e481.

Ramesh, V. P. (2017). Asymptotic notations and its applications. *Ramanujan Math Soc Math Newsl 28, no. 4*, 10-16.

Ravi, N., Scaglione, A., Kadam, S., Gentz, R., Peisert, S., Lunghino, B., . . . Shumavon, A. (2022). Differentially Private K-Means Clustering Applied to Meter Data Analysis and Synthesis. *IEEE Transactions on Smart Grid*.

Ren, X. C.-M. (2018). LoPub: high-dimensional crowdsourced data publication with local differential privacy. *IEEE Transactions on Information Forensics and Security 13, no. 9*, 2151-2166.

Rivest, R. L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM 21, no. 2*, 120-126.

Sambasivarao, L. V. (2024). Secure and Efficient Energy Trading using Homomorphic Encryption on the Green Trade Platform. *International Journal of Intelligent Systems and Applications in Engineering 12, no. 1s*, 345-360.

Saputra, Y. M. (2019). Energy demand prediction with federated learning for electric vehicle networks. *In 2019 IEEE global communications conference (GLOBECOM)* (pp. 1-6). Waikoloa: IEEE.

Sebastian Cardenas, D. J., Mukherjee, M., & Ramirez, J. E. (2023). *A review of privacy in energy applications.* Richland, WA: Pacific Northwest National Laboratory.

Singh, P. a. (2013). A survey of digital watermarking techniques, applications and attacks. *International Journal of Engineering and Innovative Technology (IJEIT) 2, no. 9*, 165-175.

Singh, R. A.-W. (2023). A privacy preserving internet of things smart healthcare financial system. *IEEE Internet of Things Journal*.

Soram, R. a. (2015). On the performance of RSA in virtual banking. *In 2015 International Symposium on Advanced Computing and Communication (ISACC)* (pp. 352-359). Silchar: IEEE.

Strepparava, D. F. (2022). Privacy and Auditability in the Local Energy Market of an Energy Community with Homomorphic Encryption. *Energies 15, no. 15*, 5386.

Su, Z. Y. (2021). Secure and efficient federated learning for smart grid with edge-cloud collaboration. *IEEE Transactions on Industrial Informatics 18, no. 2*, 1333-1344.

Sun, X. F. (2021). A survey on zero-knowledge proof in blockchain. *IEEE network 35, no. 4*, 198-205.

Syed, D. S. (2020). Privacy preservation of data-driven models in smart grids using homomorphic encryption. *Information 11, no. 7*, 357.

Taïk, A. a. (2020). Electrical load forecasting using edge computing and federated learning. *In ICC 2020-2020 IEEE international conference on communications (ICC)* (pp. 1-6). Dublin: IEEE.

Tang, X. L. (2024). Zero-knowledge proof vulnerability analysis and security auditing. *Cryptology ePrint Archive*.

Tari Labs university. (2024, June). *Rank-1 Constraint System with Application to Bulletproofs*. Retrieved from https://tlu.tarilabs.com/cryptography/rank-1

Tolpegin, V. a. (2020). Data Poisoning Attacks Against Federated Learning Systems. *25th European Symposium on Research in Computer Security, ESORICS.* Guildford, UK.

Tran, H.-Y. J. (2022). Smart meter data obfuscation with a hybrid privacy-preserving data publishing scheme without a trusted third party. *IEEE Internet of Things Journal 9, no. 17*, 16080-16095.

Vengadapurvaja, A. M. (2017). An efficient homomorphic medical image encryption algorithm for cloud storage security. *Procedia computer science 115*, 643-650.

Wang, S.-x., Chen, H.-w., Zhao, Q.-y., Guo, L.-y., Deng, X.-y., Si, W.-g., & Sun, Z.-q. (2022). Preserving scheme for user's confidential information in smart grid based on digital

watermark and asymmetric encryption. *Journal of Central South University 29, no. 2*, 726-740.
Wang, T. X. (2019). Locally private high-dimensional crowdsourced data release based on copula functions. *IEEE Transactions on Services Computing 15, no. 2*, 778-792.
Wang, T. X. (2020). A comprehensive survey on local differential privacy toward data statistics and analysis. *Sensors 20, no. 24*, 7030.
Wang, X. X. (2021). Charging station recommendation for electric vehicle based on federated learning. *In Journal of physics: Conference series, vol. 1792, no. 1* (p. 012055). IOP Publishing.
Xie, T. J. (2019). Libra: Succinct zero-knowledge proofs with optimal prover computation. *In Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference* (pp. pp. 733-764). Santa Barbara: Springer International Publishing.
Xu, W. J.-O. (2023). A Privacy-Preserving Framework Using Homomorphic Encryption for Smart Metering Systems. *Sensors 23, no. 10 (2023): 4746.*, 4746.
Yang, H. a. (2015). A secure removable visible watermarking for BTC compressed images. *Multimedia Tools and Applications*, 1725-1739.
Yang, M. T.-Y. (2023). Local differential privacy and its applications: A comprehensive survey. *Computer Standards & Interfaces*, 103827.
Yang, R. H. (2024). Advancing User Privacy in Virtual Power Plants: A Novel Zero-Knowledge Proof-Based Distributed Attribute Encryption Approach. *Electronics 13, no. 7*, 1283.
Yin, L. J. (2021). A privacy-preserving federated learning for multiparty data sharing in social IoTs. *IEEE Transactions on Network Science and Engineering 8, no. 3*, 2706-2718.
Yu, H. L. (2021). Road Distance Computation Using Homomorphic Encryption in Road Networks. *Computers, Materials & Continua 69, no. 3*.
Yuan, K. P. (2024). A Timed-Release E-Voting Scheme Based on Paillier Homomorphic Encryption. *IEEE Transactions on Sustainable Computing*.
Yucel, F. K. (2019). Efficient and privacy preserving supplier matching for electric vehicle charging. *Ad Hoc Networks 90*, 101730.
Zhang, C. Y. (2021). A survey on federated learning. *Knowledge-Based Systems 216*, 106775.
Zhang, L. J. (2023). An incremental photovoltaic power prediction method considering concept drift and privacy protection. *Applied Energy 351*, 121919.
Zhang, Z. T. (2018). CALM: Consistent adaptive local marginal for marginal release under local differential privacy. *In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (pp. 212-229). New York: Association for Computing Machinery.
Zhang, Z. Z. (2020). Benchmarking semi-supervised federated learning. *arXiv preprint arXiv:2008.11364 17, no. 1*.
Zhao, J. J. (2017). Dependent Differential Privacy for Correlated Data. *2017 IEEE Globecom Workshops (GC Wkshps)* (pp. pp. 1-7). Singapore: IEEE.
Zhou, L. A. (2024). Leveraging zero knowledge proofs for blockchain-based identity sharing: A survey of advancements, challenges and opportunities. *Journal of Information Security and Applications 80*, 103678.

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354

1-888-375-PNNL (7665)

www.pnnl.gov