Technische Universität Berlin

Information Systems Engineering
Fakultät IV
Einsteinufer 17
10587 Berlin
https://www.tu.berlin/ise

Thesis

Verifiable Data Transformations in IoT Environments using Recursive zk-SNARKs

Ramón Felipe Kühne

Matriculation Number: 456119

First Examiner: Prof. Dr.-Ing. Stefan Tai
Second Examiner: Prof. Dr. Sahin Albayrak

Supervised by
Karl Wolf
Fabian Piper

Hereby I declare that I wrote this thesis myself with the help of no more than the mentioned literature and auxiliary means.

Berlin, September 22, 2025

.....
Ramón Felipe Kühne

Declaration of Authorship

I hereby declare that I have written this thesis independently and have not used any sources or aids other than those stated. All passages taken directly or indirectly from the published or unpublished work of others have been identified as such. This thesis has not been submitted in substantially the same form to any other examination board and has not been published.

Abstract

Acknowledgements

Contents

List of Figures

List of Tables

1 Introduction

1.1 Motivation

IoT deployments such as smart home sensors, industrial monitors, and environmental sensing networks generate continuous high resolution time series data. To reduce communication and storage demands on resource constrained edge devices, this data is often aggregated in batches, for example via summation or averaging. Conventional aggregation lacks formal guarantees regarding data integrity and privacy [1], [2]. Research has shown that even coarse patterns like hourly energy usage can expose private habits [3]. Aggregation pipelines that rely on unverified local computation are susceptible to tampering or omission, which undermines trust in reported values [4]. This combination of emerging privacy threats and trust issues highlights the need for cryptographic verification mechanisms in IoT aggregation systems.

Global data production has exploded over the past decade. In 2010 approximately 2 zettabytes of data existed. By 2023 that number reached about 120 zettabytes. In 2024 it rose to approximately 147 zettabytes. Projections expect global data volume to grow further to 181 zettabytes by 2025 [5], [6]. This dramatic increase magnifies the potential impact of large scale data breaches. In the healthcare sector alone the number of breaches reported to U.S. authorities reached 725 in 2023, exposing over 133 million records [7].

The proliferation of IoT devices further accelerates data generation and increases privacy risk. Smart thermostats, smart meters, wearable health trackers, voice assistants, and environmental sensors continuously collect sensor data, often without users' full awareness. These devices shape daily life and generate intimate behavioral insights.

Because massive volumes of data are generated every moment and breaches are escalating, ensuring the integrity and confidentiality of aggregated data has become critically important. This motivates the development of cryptographic methods that can verify aggregated IoT data without compromising user privacy.

1.2 Problem Statement

The central problem of this thesis has two main aspects. First, existing aggregation methods do not provide formal integrity guarantees. In practice, users cannot confirm that published aggregates include all raw sensor readings nor detect whether any data were omitted or modified during processing. Second, although zkSNARKs allow confidential proofs of correctness, classical non recursive approaches become increasingly inefficient when used repeatedly for continuous streaming data. The core inefficiency stems from computational complexity, especially during witness generation, which can easily become a bottleneck on IoT hardware with limited resources [8]. In addition, many traditional zkSNARK protocols depend on a trusted setup and do not allow parallel processing, which limits their scalability in IoT use cases.

Recursive zkSNARKs present a promising alternative. They support proof chaining across batches, so that verification cost is amortized rather than repeated. Recent systems such as GENES demonstrate substantial improvements in proving time and verification latency through recursive proof composition. However, these improvements sometimes come with the trade off of larger overall proof sizes [9]. Likewise, Zecale demonstrates how recursive aggregation can substantially reduce verification overhead while preserving privacy in blockchain contexts [10]. Despite these theoretical advantages, the deployment of recursive zkSNARKs in constrained, privacy critical IoT environments has not yet been evaluated.

This research therefore targets a gap in current understanding by empirically establishing when recursive zkSNARKs offer a measurable advantage compared to classical zkSNARKs under realistic IoT conditions (hardware limits, privacy objectives, communication constraints). Our benchmarks compare recursive systems to non-recursive implementations using identical data and report only measured latency, memory, and proof-size results to produce actionable guidance for real-world system designers.

1.3 Research Questions

Our research is guided by the following primary questions:

1. Under which conditions is the use of recursive SNARKs beneficial?
2. What added value do recursive SNARKs provide in the context of privacy?
3. From which data volume or computational complexity onwards are recursive SNARKs more efficient?
4. Can empirical measurements on realistic IoT setups determine when recursion becomes advantageous?
5. What are the privacy-performance trade-offs in recursive vs. standard SNARK systems?

1.4 Contributions

This thesis makes the following key contributions:

1. **Nova Implementation**: Complete implementation of Nova recursive SNARKs optimized for IoT data processing
2. **Empirical Validation**: Comprehensive resource-constrained IoT evaluation
3. **Performance Analysis**: Detailed benchmarking across multiple scenarios using real measurements only
4. **Practical Guidelines**: Decision frameworks for choosing appropriate proof systems

1.5 Thesis Structure

The thesis is organized into eight chapters that map directly to the research questions and the evaluation pipeline:

1. **Introduction** (chapter 1): Motivation, problem statement, research questions, contributions, and chapter roadmap.
2. **Background** (chapter 2): Task-relevant overview of IoT aggregation privacy risks and zero-knowledge systems; emphasis on concepts required to interpret the later crossover analysis.
3. **Related Work** (chapter 3): Positioning within IoT privacy aggregation and zero-knowledge literature; highlights the gap this thesis addresses; core concepts and trade-offs of recursive vs. non-recursive zk-SNARKs relevant to this study.
4. **System Architecture** (chapter 4): End-to-end architecture and components; actors; data flow and threat model; how the project is structured and executed.
5. **Implementation** (chapter 5): Pipelines, tooling, measurement harness, reproducibility, and limitations.
6. **Empirical Results and Analysis** (chapter 6): Integrated results covering crossover validation, temporal batching, sensitivity analysis, device-level performance, and practical selection guidelines.
7. **Discussion** (chapter 7): Interpretation of findings, decision framework for practitioners, and threats to validity.
8. **Conclusion & Future Work** (chapter 8): Summary of contributions and directions for future research.

This structure keeps the narrative focused on the central objective: reporting strictly empirical advantages of recursive SNARKs in IoT settings. Each part either introduces a necessary concept, contributes a component of the methodology, or reports measured results that directly answer the research questions.

2 Background

2.1 Privacy & Data Aggregation in IoT

Resource constrained devices in Internet of Things environments collect and transmit sensor data such as temperature, power usage or motion events. Aggregating this data can reduce communication load and storage overhead, but doing so without cryptographic guarantees can compromise data integrity or privacy. A review by Ali et al [11] shows that traditional data aggregation techniques may expose raw readings and remain vulnerable to inference or tampering, especially in constrained sensor networks. Solutions such as LiPI [12] propose lightweight data masking mechanisms, but they often trade off integrity verification or depend on trusted components. There is limited research on cryptographically verifiable aggregation tailored for resource limited IoT nodes, especially when continuous privacy preservation is required.

2.2 Overview of Zero-Knowledge Proofs

A zero-knowledge proof (ZKP) lets a prover convince a verifier that a statement is true without revealing any additional information beyond its truth [13]. ZKPs exist in interactive and non-interactive forms. zk-SNARKs are non-interactive arguments of knowledge with succinct proofs; in many constructions, proof size and verifier work are sublinear—often effectively constant—in the size of the computation, though they may depend on public input size and typically require a setup [14]. zk-SNARKs have seen prominent deployments, e.g., in Zerocash/Zcash [15], [16]. zk-STARKs are transparent (no trusted setup) and hash-based; they scale well and are plausibly post-quantum, but usually incur larger proofs and higher prover costs [17]. Bulletproofs provide short non-interactive proofs without trusted setup with logarithmic proof size for certain statements (e.g., range proofs); however, verification is generally more expensive than in SNARK systems for large circuits [18]. Other families (e.g., Sonic, Plonk/Halo variants) explore different trade-offs in universality, transparency, setup, and efficiency [19]. A recent survey overviews applications and practical frameworks across domains [20].

Alternatives and PETs

Beyond zk-SNARKs, other families and PETs exist: zk-STARKs are transparent and plausibly post-quantum, but typically have larger proof sizes and higher prover costs. Differential Privacy adds calibrated noise to protect individuals but trades utility for privacy [21], [22]. Secure Multi-Party Computation offers strong privacy without a trusted party, yet incurs substantial communication/latency overhead and does not inherently provide public verifiability [23]–[25]. Given our goal (verifiable aggregation under device and bandwidth constraints), we focus on zk-SNARKs, specifically standard vs. recursive designs.

Why we do not include a non-ZK baseline

Non-cryptographic pipelines can be faster, but they reveal raw inputs or rely on trust in the computing entity, violating our privacy and integrity requirements. Without zero-knowledge proofs, a verifier cannot simultaneously ensure input confidentiality and computational correctness [13], [26]. Therefore, we compare proof systems (standard vs. recursive zk-SNARKs) rather than including a non-ZK baseline as a candidate solution.

2.3 Recursive ZKPs and Aggregation

Recursive zero knowledge proofs stack or fold multiple proofs into a single succinct result. This enables efficient and scalable verification especially in streaming or multi step computation settings where multiple sub proofs are generated.

Definition of aggregation in this thesis

We use the term *aggregation* narrowly to denote computations over sets or windows of readings that reduce raw data to concise statistics or validity results (e.g. range validation, sum/mean/median, min/max). In our scope, privacy-preserving aggregation must (i) reveal nothing about individual readings beyond what is logically implied by the output, and (ii) enable verifiers to check correctness without access to raw inputs. These requirements motivate zero-knowledge approaches [13], [26] and connect directly to our circuits and pipeline in chapter 5.

Principles and Benefits

The core idea of recursive ZKPs is to verify a proof inside another proof, thus composing multiple statements into an incrementally verifiable chain. This approach is formalized in theories such as incrementally verifiable computation and proof folding schemes. Nova introduced an efficient folding scheme that absorbs complexity into a relaxed R1CS representation, dramatically reducing per proof cost while maintaining succinct final proofs [27]. This makes recursion especially powerful when many steps must be verified sequentially.

Frameworks: Halo, Nova, Plonky2

Halo, introduced by Bowe et al in 2019, pioneered recursive SNARK designs that do not require a trusted setup. It supports cycles of elliptic curves and recursive proof composition transparently [28]. Nova builds on similar ideas through an efficient folding based proof aggregation strategy and achieves state of the art performance in proof generation and succinctness [27], [29]. Plonky2 is a zk-STARK based system optimized by Polygon Zero for recursive workloads. It uses custom gates and deep arithmetic constraints to enable recursion at scale with high proving speed [30]–[33]. All three systems allow continual chaining of proofs and compression into a single final proof, reducing verification overhead in multi step or streaming use cases.

3 Related Work

3.1 IoT Privacy and Data Aggregation

The overlap between IoT privacy preservation and data aggregation has been extensively studied. Traditional approaches to IoT data aggregation often sacrifice privacy for efficiency, creating vulnerabilities in smart home and industrial deployments [1], [2].

Privacy-Preserving IoT Aggregation

Early work by Ali et al. demonstrated that conventional aggregation techniques expose raw sensor readings to inference attacks [11]. Solutions such as LiPI propose lightweight obfuscation mechanisms but often trade off integrity verification or depend on trusted components [12].

Recent advances in differential privacy for IoT have shown promise but struggle with the continuous, high-frequency nature of sensor data. The challenge lies in balancing privacy preservation with the computational and energy constraints of IoT devices. Prior work rarely offers end-to-end, verifiable aggregation with measured crossover points under resource constraints; our study fills this empirical gap by reporting only measured results and explicit crossover regimes.

Critical positioning

Compared to prior surveys and systems, our contribution is explicitly empirical and hardware-aware: (i) we benchmark standard vs. recursive zk-SNARKs on identical logic and data under matched resource limits enforced via Docker (CPU/RAM constraints), and (ii) we report time/size crossovers under steady-state operation. Many prior works emphasize protocol design or transparency properties (e.g., STARKs) without quantifying when recursion is advantageous under constrained compute and memory. Our results provide that missing, practical boundary.

3.2 Recursive vs. Non-Recursive zk-SNARKs in Resource-Constrained Environments

Fundamentals: Difference Between Recursive and Non-Recursive zk-SNARKs

zk-SNARKs (Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge) allow one to prove the correctness of a computation using a short cryptographic proof without revealing the underlying data. A non-recursive zk-SNARK refers to a single proof for a specific computation or statement. In contrast, recursive zk-SNARKs allow multiple proofs or computation steps to be composed into each other. In recursion, the output of one zk-SNARK is used as part of the input for the next, resulting in a single final proof that attests to

the correctness of all intermediate computations [34]. This is also known as incrementally verifiable computation (IVC): the prover produces a proof for each computation step that confirms both the correctness of that step and that the previous step was correctly verified [35]. Through this composition, iterative or sequential computations can be securely chained.

A well-known example of recursive zk-SNARKs is Nova, which is based on a folding scheme. Nova folds a long computation into an ongoing recursive proof and only generates the final zk-SNARK at the end [8]. As a result, the expensive zk-SNARK generation occurs only once—regardless of how many steps were involved in the computation. Systems such as Halo or Nova have demonstrated that recursive zk-SNARKs can be built without a trusted setup, making them suitable for real-world applications [8].

Efficiency, Computation Cost, and Latency

The primary efficiency difference lies in the trade-off between proof generation and verification. In non-recursive zk-SNARKs, generating a single proof is expensive, but verifying that proof is very fast (often milliseconds). However, if multiple zk-SNARKs must be verified (e.g., many individual proofs), the overall verification time scales linearly. Recursive zk-SNARKs aim to drastically reduce this verification overhead by aggregating all claims into a single proof [34]. Thus, the final verification time remains essentially constant, regardless of the number of individual steps or proofs involved.

On the proving side, recursive SNARKs introduce some overhead, since each new proof must verify the previous one, increasing the number of constraints. In traditional constructions (e.g., Groth16), verifying a SNARK inside a SNARK was costly. Modern systems like Nova optimize this by delaying the expensive zk-SNARK compression to the end [8]. Nova works in two stages: it first builds an ongoing recursive proof and then applies a final zk-SNARK compression. This final step incurs a fixed cost, regardless of how many steps were folded in. Hence, the final verification time remains constant, while the proof generation time increases roughly linearly with the number of steps. Latency may increase moderately, since the system waits until the end to compress the accumulated proofs.

Proof size is another major advantage. While a typical Groth16 proof is constant in size, producing many individual proofs results in linear growth in storage or transmission. Recursive SNARKs produce one final compact proof whose size is largely independent of the number of inputs [34].

Scalability

Recursive zk-SNARKs are most beneficial when dealing with large-scale computations or proof aggregation. For small or one-time computations, a single non-recursive proof is often more efficient, as the recursive overhead may not be justified.

Empirical studies indicate that even at modest batch sizes (a few dozen proofs), recursion can become advantageous. For example, in a decentralized IoT setting, Nova required only ~3.6 seconds to aggregate and verify 10 digital signatures, whereas a non-recursive method using Risc0 took ~369 seconds—over 100× slower [36]. The gap grows with more inputs. Another study showed that Nova could verify 100 signatures in 7.1 seconds, whereas a previous method based on homomorphic encryption and ECDSA took over 50 seconds to verify just 64 signatures [36]. These results suggest that at batch sizes of a few dozen, recursive approaches can already be significantly more efficient.

Moreover, recursion reduces distributed verification overhead. Without recursion, each verifier must check all proofs. With recursion, only a single final proof needs to be verified. This makes the per-claim verification time negligible, since a constant cost is amortized over many claims [36]. The load is shifted from weak verifiers (e.g., IoT devices or smart contracts) to a single strong prover.

Use in IoT and Smart-Home Scenarios

IoT and smart-home environments impose strict constraints: sensors and embedded devices often have limited processing power, memory, and energy. zk-SNARK generation is typically too expensive to perform locally [36]. Even verification can overwhelm constrained devices. Therefore, many architectures follow a layered model with edge servers.

In this setup, IoT devices only collect and sign data. They then forward it to a nearby edge aggregator, which performs proof generation and aggregation [36]. Only the final proof or its hash is sent to a blockchain or central verifier. This eliminates the need for IoT devices to generate or verify SNARKs, saving energy and bandwidth.

Recursive zk-SNARKs are ideal for such scenarios, as they can aggregate continuous sensor streams into an ongoing proof. For instance, Nova has been used to aggregate and verify 100 sensor signatures into a single proof suitable for on-chain verification [36]. Verifying this proof took only ~0.06 s per signature (i.e., ~6 s total), even for low-powered verifiers.

Beyond signature verification, recursive SNARKs can prove compliance with rules over long periods, such as "no sensor exceeded a threshold for the past hour." This streaming proof model allows incremental updates and compact final validation, ideal for constrained environments [8].

Studies have even demonstrated recursive zk-SNARKs in advanced tasks like federated learning: each local training round and the global aggregation step are provably verified using Nova. In one setup, the global model proof took ~81 seconds to generate and ~0.6 seconds to verify [35]. This shows that the cost is mostly on the proving side, which can be offloaded to strong devices.

Summary and Implications for Architecture

Recursive zk-SNARKs offer compelling benefits for scaling zero-knowledge applications in IoT scenarios. They enable aggregation of multiple computations or data streams into a single compact proof, which reduces memory, bandwidth, and verification cost—key concerns in resource-constrained environments. Our system architecture (chapter 4) therefore places proving at an edge aggregator, defines batching policies that drive into empirically observed crossover regimes, and adopts metrics (proof time, verification time, proof size, and device load) that operationalize these trade-offs. The following chapter translates these implications into a concrete architecture and methodology and specifies the evaluation setup used to validate them in practice.

4 System Architecture

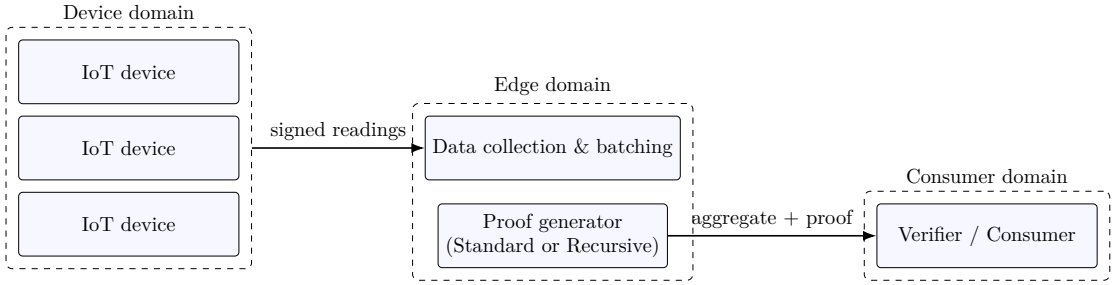

Figure 4.1: System Architecture and logic

Component terminology (Figure 4.1).

- **Domain**: *Device domain* groups IoT devices, *Edge domain* hosts data collection/batching and proving, *Consumer domain* contains verifiers and consuming services.
- **Signed readings**: Sensor measurements signed by the originating device (with timestamp/ID); ensures authenticity and integrity of inputs to aggregation.
- **Data collection & batching**: Edge component that verifies signatures, deduplicates/orders events, buffers them, and forms fixed windows/batches as circuit inputs.
- **Proof generator (Standard or Recursive)**: Cryptographic stage creating a zk-proof that the published aggregate(s) were computed correctly from the private inputs; implemented once with non-recursive SNARKs and once with recursion.
- **Aggregate + proof**: Public outputs (aggregates) together with the corresponding zero-knowledge proof delivered to consumers.
- **Verifier / Consumer**: Validates proofs and uses the aggregates (dashboards, analytics, policy, billing, etc.).

This chapter establishes the evaluation framework used to compare standard and recursive zk-SNARKs in IoT environments and defines the system components at a technology-agnostic level. The architecture follows an edge-centric pattern [37]. ould be detected with overwhelming probability.

4.1 Privacy Definition

We define privacy in this work as the non-disclosure of individual sensor readings and intermediate computation states. A verifier learns only: (i) declared aggregates, and (ii) that

these aggregates are consistent with some private inputs satisfying the circuit constraints. By zero-knowledge, the proof transcript is simulatable without access to raw inputs and reveals no information beyond statement validity [13], [26], [38].

5 Implementation

5.1 Environment and Hardware Profile

We executed all experiments under a resource-constrained profile intended to simulate realistic IoT edge environments. Specifically, we used Docker containers with enforced limits on CPU, memory, and swap space to approximate the hardware characteristics of IoT-devices, such as Raspberry Pi's.

- **CPU**: fixed 0.5 logical core
- **RAM**: fixed 1 GB
- **Swap**: 1 GB (additional virtual memory)
- **Storage**: SSD, same filesystem for fair I/O caching behavior

These settings are chosen based on typical Raspberry Pi-class device specifications, which often provide 1-4 GB RAM and single-to-quad-core CPUs in edge deployments. For example, Gupta & Nahrstedt [39] empirically evaluate similar IoT devices and show that Docker containers on hardware with 1 GB RAM suffer measurable overheads in latency and I/O under constrained CPU/RAM settings.

Host platform (for emulation):

Component	Specification
CPU	AMD Ryzen 7 7800X3D (8 cores, 16 threads; base 4.2 GHz)
RAM	32 GB DDR5 @ 6000 MT/s (2 × DIMM)
GPU	NVIDIA GeForce RTX 4070 SUPER (12 GB VRAM)
Virtualization	Windows 11 + WSL2 (Ubuntu kernel 6.6.87.2)

Table 5.1: Host platform used to simulate IoT-like environments under resource constraints via Docker/cgroups.

6 Results and Analysis

7 Discussion

Alternatives and PETs: positioning. Complementary privacy-enhancing approaches offer different trade-offs for IoT aggregation. Differential Privacy protects individuals by adding calibrated noise but trades utility for privacy and does not provide integrity of computation [21], [22]. Secure Multiparty Computation can compute over distributed inputs without a trusted party but typically incurs higher latency and communication overhead that challenge constrained devices [23]–[25]. Trusted Execution Environments provide hardware-backed isolation but introduce additional trust assumptions and potential side-channel risks **costan2016sgx**. Our evaluation therefore focuses on zk-SNARKs (Standard vs. Nova) to obtain end-to-end verifiability with confidentiality and constant-size verification artifacts; PETs are discussed here to clarify scope and to motivate future comparative work.

8 Conclusion & Future Work

Bibliography

[1] J. Kua, M. B. Hossain, I. Natgunanathan, and Y. Xiang, "Privacy Preservation in Smart Meters: Current Status, Challenges and Future Directions," en, *Sensors*, vol. 23, no. 7, p. 3697, Jan. 2023, Number: 7 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 1424-8220. DOI: 10.3390/s23073697. [Online]. Available: https://www.mdpi.com/1424-8220/23/7/3697 (visited on 05/18/2025).

[2] F. Kabir, A. Qureshi, and D. Megias, *A Study on Privacy-Preserving Data Aggregation Techniques for Secure Smart Metering System*, eng. Apr. 2021, Accepted: 2024-11-15T08:32:37Z, ISBN: 978-84-09-29150-2. [Online]. Available: https://openaccess.uoc.edu/handle/10609/151535 (visited on 05/18/2025).

[3] K. J. Müller, "Gewinnung von Verhaltensprofilen am intelligenten Stromzähler," de, *Datenschutz und Datensicherheit - DuD*, vol. 34, no. 6, pp. 359–364, Jun. 2010, ISSN: 1862-2607. DOI: 10.1007/s11623-010-0107-2. [Online]. Available: https://doi.org/10.1007/s11623-010-0107-2 (visited on 04/06/2025).

[4] J.-M. Bohli, C. Sorge, and O. Ugus, "A Privacy Model for Smart Metering," in *2010 IEEE International Conference on Communications Workshops*, ISSN: 2164-7038, May 2010, pp. 1–5. DOI: 10.1109/ICCW.2010.5503916. [Online]. Available: https://ieeexplore.ieee.org/document/5503916/ (visited on 04/06/2025).

[5] *IDC: Expect 175 zettabytes of data worldwide by 2025*, en. [Online]. Available: https://www.networkworld.com/article/966746/idc-expect-175-zettabytes-of-data-worldwide-by-2025.html (visited on 07/25/2025).

[6] *How much data is generated every day?* [Online]. Available: https://soax.com/research/data-generated-per-day?utm_source=chatgpt.com (visited on 07/25/2025).

[7] S. Alder, *December 2023 Healthcare Data Breach Report*, en-US, Jan. 2024. [Online]. Available: https://www.hipaajournal.com/december-2023-healthcare-data-breach-report/ (visited on 07/25/2025).

[8] M. El-Hajj and B. Oude Roelink, "Evaluating the Efficiency of zk-SNARK, zk-STARK, and Bulletproof in Real-World Scenarios: A Benchmark Study," en, *Information*, vol. 15, no. 8, p. 463, Aug. 2024, Number: 8 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2078-2489. DOI: 10.3390/info15080463. [Online]. Available: https://www.mdpi.com/2078-2489/15/8/463 (visited on 07/26/2025).

[9] J. Liu, L. Guo, and T. Kang, "GENES: An Efficient Recursive zk-SNARK and Its Novel Application in Blockchain," en, *Electronics*, vol. 14, no. 3, p. 492, Jan. 2025, Number: 3 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2079-9292. DOI: 10.3390/electronics14030492. [Online]. Available: https://www.mdpi.com/2079-9292/14/3/492 (visited on 04/18/2025).

[10] A. Rondelet, *Zecale: Reconciling Privacy and Scalability on Ethereum*, arXiv:2008.05958 [cs], Oct. 2020. DOI: 10.48550/arXiv.2008.05958. [Online]. Available: http://arxiv.org/abs/2008.05958 (visited on 07/26/2025).

[11] I. Ali, S. Sabir, and E. Khan, *Privacy-preserving data aggregation in resource-constrained sensor nodes in Internet of Things: A review*, arXiv:1812.04216 [cs], Dec. 2018. DOI: 10.48550/arXiv.1812.04216. [Online]. Available: http://arxiv.org/abs/1812.04216 (visited on 07/26/2025).

[12] H. Goyal, K. Kodali, and S. Saha, *LiPI: Lightweight Privacy-Preserving Data Aggregation in IoT*, arXiv:2207.12197 [cs], Jul. 2022. DOI: 10.48550/arXiv.2207.12197. [Online]. Available: http://arxiv.org/abs/2207.12197 (visited on 07/26/2025).

[13] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof-systems," en, in *Proceedings of the seventeenth annual ACM symposium on Theory of computing - STOC '85*, Providence, Rhode Island, United States: ACM Press, 1985, pp. 291–304, ISBN: 978-0-89791-151-1. DOI: 10.1145/22145.22178. [Online]. Available: http://portal.acm.org/citation.cfm?doid=22145.22178 (visited on 03/09/2025).

[14] A. Nitulescu, "Zk-SNARKs: A Gentle Introduction," en,

[15] E. Ben Sasson, A. Chiesa, C. Garman, *et al.*, "Zerocash: Decentralized Anonymous Payments from Bitcoin," in *2014 IEEE Symposium on Security and Privacy*, ISSN: 2375-1207, May 2014, pp. 459–474. DOI: 10.1109/SP.2014.36. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/6956581 (visited on 09/20/2025).

[16] D.-E. Hopwood, S. Bowe, T. Hornby, and N. Wilcox, "Zcash Protocol Specification, Version 2025.6.0-79-g7d61ca [NU6.1 proposal],"

[17] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev, "Scalable Zero Knowledge with No Trusted Setup," en, in *Advances in Cryptology – CRYPTO 2019*, A. Boldyreva and D. Micciancio, Eds., vol. 11694, Series Title: Lecture Notes in Computer Science, Cham: Springer International Publishing, 2019, pp. 701–732, ISBN: 978-3-030-26953-1 978-3-030-26954-8. DOI: 10.1007/978-3-030-26954-8_23. [Online]. Available: https://link.springer.com/10.1007/978-3-030-26954-8_23 (visited on 09/20/2025).

[18] *Bulletproofs | Stanford Applied Crypto Group*. [Online]. Available: https://crypto.stanford.edu/bulletproofs/?utm_source=chatgpt.com (visited on 09/20/2025).

[19] M. Maller, S. Bowe, M. Kohlweiss, and S. Meiklejohn, "Sonic: Zero-Knowledge SNARKs from Linear-Size Universal and Updatable Structured Reference Strings," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '19, New York, NY, USA: Association for Computing Machinery, Nov. 2019, pp. 2111–2128, ISBN: 978-1-4503-6747-9. DOI: 10.1145/3319535.3339817. [Online]. Available: https://doi.org/10.1145/3319535.3339817 (visited on 09/20/2025).

[20] R. Lavin, X. Liu, H. Mohanty, L. Norman, G. Zaarour, and B. Krishnamachari, *A Survey on the Applications of Zero-Knowledge Proofs*, arXiv:2408.00243 [cs] version: 1, Aug. 2024. DOI: 10.48550/arXiv.2408.00243. [Online]. Available: http://arxiv.org/abs/2408.00243 (visited on 07/26/2025).

[21] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," en-US, Mar. 2006, pp. 265–284. [Online]. Available: https://www.microsoft.com/en-us/research/publication/calibrating-noise-to-sensitivity-in-private-data-analysis/ (visited on 09/15/2025).

[22] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," en, *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3-4, pp. 211–407, 2013, ISSN: 1551-305X, 1551-3068. DOI: 10.1561/0400000042. [Online]. Available: http://www.nowpublishers.com/articles/foundations-and-trends-in-theoretical-computer-science/TCS-042 (visited on 09/15/2025).

[23] A. C.-C. Yao, "How to generate and exchange secrets," in *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, ISSN: 0272-5428, Oct. 1986, pp. 162–167. DOI: 10.1109/SFCS.1986.25. [Online]. Available: https://ieeexplore.ieee.org/document/4568207 (visited on 09/15/2025).

[24] O. Goldreich, S. Micali, and A. Wigderson, "How to play ANY mental game," in *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, ser. STOC '87, New York, NY, USA: Association for Computing Machinery, Jan. 1987, pp. 218–229, ISBN: 978-0-89791-221-1. DOI: 10.1145/28395.28420. [Online]. Available: https://dl.acm.org/doi/10.1145/28395.28420 (visited on 09/15/2025).

[25] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, "Multiparty Computation from Somewhat Homomorphic Encryption," en, in *Advances in Cryptology – CRYPTO 2012*, R. Safavi-Naini and R. Canetti, Eds., Berlin, Heidelberg: Springer, 2012, pp. 643–662, ISBN: 978-3-642-32009-5. DOI: 10.1007/978-3-642-32009-5_38.

[26] J. Katz and Y. Lindell, *Introduction to Modern Cryptography: Principles and Protocols*. New York: Chapman and Hall/CRC, Aug. 2007, ISBN: 978-0-429-14380-9. DOI: 10.1201/9781420010756.

[27] A. Bassa, *Intro to Nova & ZK folding schemes: Halo and accumulation*, en-US, Aug. 2023. [Online]. Available: https://veridise.com/blog/learn-blockchain/intro-to-nova-zk-folding-schemes-halo-and-accumulation/ (visited on 07/26/2025).

[28] S. Bowe, J. Grigg, and D. Hopwood, "Halo: Recursive Proof Composition without a Trusted Setup," en,

[29] *The Pantheon of Zero Knowledge Proof Development Frameworks (Updated!)* en-US, Aug. 2023. [Online]. Available: https://blog.celer.network/2023/08/04/the-pantheon-of-zero-knowledge-proof-development-frameworks/ (visited on 07/26/2025).

[30] *The Plonky2 Recursive Zero-Knowledge Proof*, en-US. [Online]. Available: https://www.zkm.io/blog/the-plonky2-recursive-zero-knowledge-proof (visited on 07/26/2025).

[31] *Analysis of The Plonky2 Protocol*, en-US. [Online]. Available: https://www.zkm.io/blog/analysis-of-the-plonky2-protocol (visited on 07/26/2025).

[32] *Introducing Plonky2*, en. [Online]. Available: https://polygon.technology/blog/introducing-plonky2 (visited on 07/20/2025).

[33] *Maya ZK Blog*. [Online]. Available: https://www.maya-zk.com/blog/proof-aggregation?utm_source=chatgpt.com (visited on 07/26/2025).

[34] R. Innovation, *Blockchain Scalability Guide 2024: Layer 2 Solutions*, en-US. [Online]. Available: https://www.rapidinnovation.io/post/blockchain-scalability-solutions-layer-2-and-beyond (visited on 08/05/2025).

[35] A. A. Bellachia, M. A. Bouchiha, Y. Ghamri-Doudane, and M. Rabah, *VerifBFL: Leveraging zk-SNARKs for A Verifiable Blockchained Federated Learning*, arXiv:2501.04319 [cs], Jan. 2025. DOI: 10.48550/arXiv.2501.04319. [Online]. Available: http://arxiv.org/abs/2501.04319 (visited on 08/05/2025).

[36] J. Bojič Burgos and M. Pustišek, "Decentralized IoT Data Authentication with Signature Aggregation," en, *Sensors*, vol. 24, no. 3, p. 1037, Jan. 2024, Number: 3 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 1424-8220. DOI: 10.3390/s24031037. [Online]. Available: https://www.mdpi.com/1424-8220/24/3/1037 (visited on 06/03/2025).

[37] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct. 2016, ISSN: 2327-4662. DOI: 10.1109/JIOT.2016.2579198. [Online]. Available: https://ieeexplore.ieee.org/document/7488250 (visited on 09/15/2025).

[38] O. Goldreich, *Foundations of Cryptography: Volume 2, Basic Applications*, en. Cambridge University Press, 2001, Google-Books-ID: tfzM9d_jnxwC, ISBN: 978-0-521-83084-3.

[39] R. Gupta and K. Nahrstedt, *Performance Characterization of Containers in Edge Computing*, arXiv:2505.02082 [cs], May 2025. DOI: 10.48550/arXiv.2505.02082. [Online]. Available: http://arxiv.org/abs/2505.02082 (visited on 09/22/2025).