Verifiable Data Transformations in IoT Environments using Recursive zk-SNARKs

Ramón Felipe Kühne
Student number: 456119

Examiners
First Examiner: Prof. Dr.-Ing. Stefan Tai
Second Examiner: Axel Küpper (SNET) or Stefan Schmid (INET)

Supervisors
Karl Christoph Wolf, Fabian Piper

Department of Information Systems Engineering (ISE), TU Berlin

Introduction

Internet of Things (IoT) deployments such as smart home sensors, industrial monitors, and environmental sensing networks produce continuous high-resolution time-series data [1]. Aggregating this data in periodic batches—for example by summation or averaging—reduces communication and storage demands on resource-constrained devices but does not provide formal guarantees that the published aggregates correctly reflect all raw readings or that individual measurements remain confidential [2]. In practice, simple aggregation schemes have been shown to leak sensitive patterns: Müller demonstrated that even hourly load profiles in smart homes can reveal user behavior [3], and such schemes remain vulnerable to tampering or data omission without verifiable proofs [4].

Consider a typical smart home equipped with devices such as thermostats, motion sensors, smart meters, and voice assistants. These devices continuously collect detailed sensor readings, which, when aggregated over a defined interval, can reveal intimate insights into the occupants' daily routines and personal habits [5]. For example, consistent energy consumption patterns reported by a smart meter over a given interval can indicate when residents are at home, asleep, or away, posing significant privacy risks [1], [6]. The problem is not that the physical meter is unreliable, but that the intermediary step (Local Aggregator) is a black box whose internal rules the user cannot control. Because the aggregation algorithm remains a black box, simple batch aggregation fails to ensure integrity and privacy simultaneously. Users generally trust individual measurement devices—such as a

certified power meter—because they can directly observe the meter readout and verify its calibration. However, once these raw readings are collected by a local data aggregator and forwarded to an external electricity provider or cloud platform, the occupant loses visibility into which measurements were used or whether any values were omitted or altered [7], [8]. This creates a trust gap: the resident sees each raw reading displayed in real time on the meter itself, but cannot confirm that the local data aggregator correctly summed all readings to produce the reported aggregate over the selected interval. Because the aggregation algorithm and data filtering steps remain non-transparent, there is a risk that data aggregators or cloud services could drop peak readings or adjust values to underreport usage, and the occupant has no mechanism to detect such manipulation. Economic incentives further exacerbate this issue: an electricity provider might underreport consumption to avoid peak-demand charges or penalties, since altering raw meter values at the device itself would be immediately noticed, but tampering during aggregation could go undetected [9]. Consequently, users lack any effective way to verify that the published aggregate equals the sum of all raw measurements, making simple batch aggregation insufficient for both integrity and privacy.

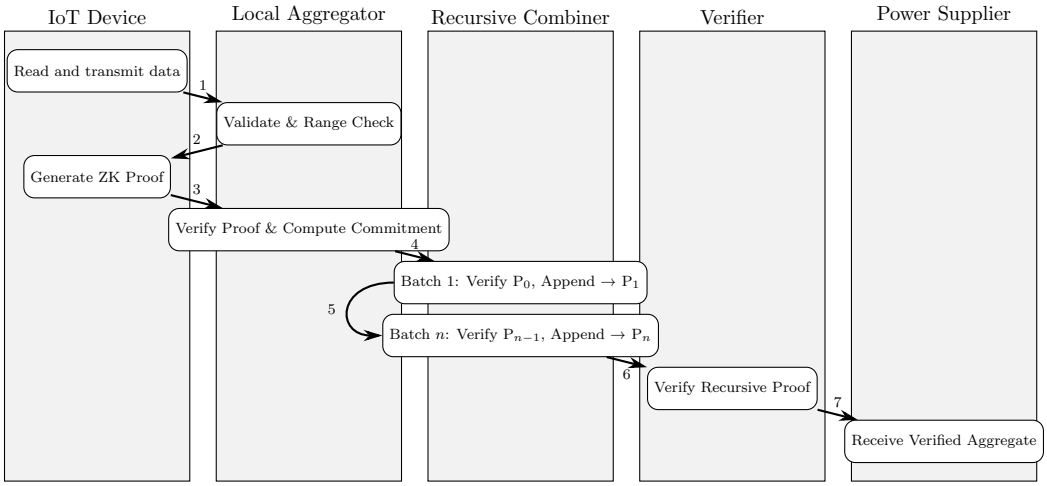
Zero-Knowledge Proofs (ZKPs) can bridge this trust gap by allowing devices to cryptographically prove that aggregated data falls within expected parameters without revealing individual sensor readings [10]. By generating succinct proofs of correct aggregation, ZKPs ensure that neither the local data aggregator nor the external electricity provider can manipulate or misreport data, while still hiding raw measurements [10]. In this way, ZKPs drastically reduce the risk of data leaks by limiting the exposure of sensitive information and protect against data manipulation, since generating valid proofs for falsified data is computationally infeasible for attackers [5]. However, implementing ZKPs in IoT environments is technically challenging due to resource constraints: many cryptographic operations associated with ZKPs are computationally expensive. zkSNARKs address these limitations effectively by generating compact proofs and requiring constant-time verification independent of the computation complexity [11], making them ideal for scenarios where edge devices must regularly send proofs to a local data aggregator or an external electricity provider. Alternatives such as interactive protocols or zkSTARKs have significant drawbacks, including larger proofs, higher verification costs, or increased communication overhead [12], [13].

Recursive zkSNARKs are particularly suited for continuous IoT data processing. Traditional methods would either generate separate proofs for every interval—leading to inefficiencies in managing numerous proofs—or create impractically large single proofs. Recursive composition elegantly addresses this by continuously combining smaller individual proofs into a single, compact end-proof [14]. For example, a resource-constrained de-

vice can periodically generate incremental proofs, chaining them together to ensure ongoing integrity verification; techniques like Nova demonstrate substantial performance advantages by efficiently aggregating multiple data verifications into a concise final proof [15]. Recursive zkSNARKs thus provide scalable, resource-efficient verification suitable for IoT devices handling continuous data streams.

To fill this gap, we propose a general recursive zkSNARK-based architecture for privacy-preserving batch aggregation in IoT environments. Our design guarantees integrity by cryptographically binding each batch proof to all prior summaries so that any missing or altered reading invalidates the chain, confidentiality by revealing only the succinct recursive proof and hiding raw sensor values, and resource efficiency by optimizing circuits and proof composition for low-power IoT hardware. We implement and evaluate our prototype in a smart-home setting where edge devices collect measurements at fixed intervals and periodically submit aggregate proofs. The smart-home scenario offers a readily accessible and representative environment for development and testing, ensuring realistic data patterns and deployment conditions. This modular, application-independent framework can be readily adapted to industrial telemetry, environmental monitoring, and other continuous-streaming scenarios.

System Architecture



Our architecture includes five distinct participants, as illustrated in the diagram above. *IoT Devices* continuously collect fine-grained sensor readings over a predefined time interval and transmit each raw measurement to the *Local Aggregator*. The *Local Aggregator* validates that the received measurement lies within predefined bounds; once validation succeeds, the IoT Device generates a zkSNARK proof [11] that its interval summary (for

example, total energy consumed) correctly reflects all private readings and satisfies the agreed constraints. The *Local Aggregator* then verifies this proof and computes a short commitment encoding the validated batch summary, which it forwards to the next stage. The *Recursive Combiner* verifies each incoming commitment against the previous recursive proof and integrates the new commitment into an updated recursive proof, ensuring that every batch remains cryptographically linked to all prior summaries [14], [16], [17]. Finally, the *Verifier* checks the complete recursive proof in constant time [15]. Because this proof remains succinct, verification can be executed with minimal overhead. The *Power Supplier* then receives verified aggregate meaning it obtains only the compact, publicly verifiable proof (not any raw readings) and can trust that the published aggregate truly reflects every device's data over all intervals. By cryptographically binding each batch summary to its entire history, any attempt to drop or alter a reading anywhere in the pipeline will break verification, thus ensuring end-to-end data integrity and preserving user privacy.

Applicability to Blockchain Contexts

Although our primary focus is on IoT environments where a local aggregator or a centralized verifier checks proofs, this recursive zkSNARK architecture can be adapted to blockchain deployments with minimal changes. In a blockchain scenario, the public ledger takes on the role of the Verifier. Instead of sending aggregate proofs to a central backend, each edge device or a designated aggregator node submits its succinct recursive proof as a transaction on the chain. A smart contract running on that chain then executes the same zkSNARK verification steps, confirming that each batch summary correctly incorporates all prior intervals without revealing any raw sensor readings. Because zkSNARK proofs remain constant in size, the on-chain gas costs stay fixed regardless of the number of intervals or devices involved. As a result, IoT devices can rely on the blockchain's immutability and consensus rules for both data integrity and ordering. This means that the privacy-preserving and integrity-ensuring properties we designed for centralized settings carry over directly to fully decentralized architectures.

Related Work

Existing work on privacy-preserving aggregation in IoT environments spans from early models in smart metering to more recent zkSNARK-based systems. Initial approaches used centralized aggregators or differential privacy mechanisms to mask individual device readings, yet these lacked formal integrity guarantees and remained vulnerable to statistical inference [3], [4]. More advanced protocols, such as the secret-sharing scheme by Kursawe et al. [7] and the on-device verification model by Rial and Danezis [8],

introduced stronger privacy controls, but still required interactive proofs for each batch and were not well suited for large-scale or autonomous IoT deployments. Blockchain-based architectures like ZGridBC demonstrated how zkSNARKs could ensure privacy and integrity in energy data markets, though they relied heavily on fixed-size batch commitments and incurred high on-chain costs [18]. Meanwhile, off-chain solutions such as the netting protocols by Eberhardt et al. leveraged zkSNARKs via the ZoKrates toolkit to reduce settlement overhead, but generated a separate proof for every aggregation interval and did not support recursive composition [19], [20].

Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge (zkSNARKs) are cryptographic proof systems that allow a prover to demonstrate the validity of a statement without revealing any underlying information. The term *succinct* indicates that these proofs are extremely compact, typically requiring minimal computational resources for verification, often just milliseconds. *Non-interactive* means the proof is provided in a single message, eliminating the need for multiple rounds of communication between prover and verifier [21], [22]. Originally popularized through applications in blockchain technologies like Zcash [14], zkSNARKs enable transactions and computations to be verified cryptographically without exposing sensitive details, making them highly suitable for privacy-sensitive IoT scenarios.

Recursive SNARKs, as introduced by Bitansky et al. in the form of proof-carrying data [23], and later extended by Bowe et al. in the Halo framework [16], offer constant-size recursive verification but have been primarily applied to static or blockchain data contexts. Deng and Du's zkTree system similarly enables on-chain verification of hierarchically composed proofs but is tailored to ledger consistency [17]. The GENES architecture optimizes recursive proofs for throughput and memory in blockchain use cases, but makes assumptions that are incompatible with dynamic sensor data streams [24]. Finally, recent embedded zkSNARK applications such as zRA and ZEKRA extend zero-knowledge techniques to firmware and control-flow attestation [25], [26], yet operate on static codebases and lack mechanisms for continuous batch processing of input data.

While these systems demonstrate important building blocks, none of them provide a reusable, streaming-compatible zkSNARKs-based architecture specifically designed for resource-constrained IoT scenarios. In contrast, this thesis proposes a general framework that combines recursive proof composition, efficient circuit logic, and modular ZoKrates implementation to support unbounded IoT data aggregation with minimal overhead. The approach is distinct in that it avoids blockchain dependencies, supports ongoing recursive updates, and prioritizes adaptability across diverse IoT domains.

Research Question

In this thesis, we propose a recursive zkSNARKs-based architecture specifically designed for privacy-preserving batch processing of streaming IoT data. The following research question will be addressed:

How can a recursive zkSNARKs-based architecture be designed to ensure the confidentiality and integrity of batched IoT data streams while remaining efficient and practical for resource-constrained devices?

Contributions

This thesis makes the following contributions:

- **Systematic design of a recursive zkSNARKs-based architecture** for privacy-preserving aggregation of time-series IoT data, developed from a comprehensive analysis of existing proof systems and their limitations in dynamic, streaming environments.
- **Implementation of a modular prototype in a smart-home environment** using the ZoKrates framework that supports recursive proof generation over multiple time intervals, ensuring confidentiality and integrity without exposing raw sensor data.
- **Demonstration of generalizability** through an application-independent circuit design and a transferable workflow.
- **Evaluation of performance, privacy and scalability** using Docker-based deployments that emulate constrained devices, validating the architecture's practical feasibility through quantitative benchmarks and security analysis.

Methodology

The research proceeds in two primary phases. In the first phase, existing privacy-preserving aggregation schemes, recursive zkSNARK constructions, and relevant IoT constraints are systematically reviewed. Particular attention is paid to the limitations of current approaches—such as the lack of recursive proof chaining, reliance on rigid batch structures, or limited adaptability to real-world data streams. Based on this review, a generalized, application-independent system architecture will be developed. This architecture defines a modular proof composition process designed for continuous sensor data and optimized for use on devices with limited computational and memory resources.

In the second phase, this architecture will be implemented as a recursive zkSNARK-based proof system using the ZoKrates framework. A key feature is the recursive aggregation mechanism, which securely connects multiple data batches into a single, verifiable proof. This approach aims to reduce communication and verification overhead while ensuring that raw input data remains confidential. The underlying circuit design will implement core logic such as data aggregation, range validation, and recursive linkage between proof intervals. We will demonstrate the prototype in a smart-home environment—where edge devices generate minute-level sensor readings and periodically report aggregates—to illustrate practical feasibility. Although demonstrated through this representative IoT scenario, the design is intended to be transferable across domains, including industrial telemetry, environmental monitoring, or smart infrastructure.

Evaluation

The evaluation of the proposed system will be carried out on several levels. To simulate realistic IoT environments, all components will be deployed using Docker containers that emulate resource-constrained edge devices. This setup allows for reproducible benchmarks while maintaining close alignment with typical embedded system constraints.

First, the system's privacy-preserving properties will be evaluated through a comparison of data observability before and after aggregation, using methodology inspired by Müller's analysis of behavioral inference from load profiles [3].

Second, a performance evaluation will assess the feasibility of recursive zkSNARK usage in constrained settings. Metrics such as proof generation time, verification latency, memory consumption, and proof size will be recorded across different batch configurations. These measurements follow established practices in prior work on zero-knowledge proof performance, such as Samudrala et al. [24].

Third, a security analysis will validate the system's resistance to manipulation by ensuring that any single malformed or inconsistent input invalidates the recursive proof chain. This guarantees the integrity of the entire aggregation process and enforces tamper-resilience at every step.

Lastly, we assess real-world applicability by analyzing the system's communication overhead, maintenance complexity, and scalability in representative IoT use cases. Although tested through one specific scenario, the architecture will be shown to generalize to broader classes of time-series aggregation problems.

References

[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] Z. Chen, Y. Jiang, X. Song, and L. Chen, "A Survey on Zero-Knowledge Authentication for Internet of Things", en, *Electronics*, vol. 12, no. 5, p. 1145, Jan. 2023, Number: 5 Publisher: Multidisciplinary Digital Publishing Institute, ISSN: 2079-9292. DOI: 10.3390/electronics12051145. [Online]. Available: https://www.mdpi.com/2079-9292/12/5/1145 (visited on 03/09/2025).

[6] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin, "Private memoirs of a smart meter", in *Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building*, ser. BuildSys '10, New York, NY, USA: Association for Computing Machinery, Nov. 2010, pp. 61–66, ISBN: 978-1-4503-0458-0. DOI: 10.1145/1878431.1878446. [Online]. Available: https://dl.acm.org/doi/10.1145/1878431.1878446 (visited on 04/18/2025).

[7] K. Kursawe, G. Danezis, and M. Kohlweiss, "Privacy-Friendly Aggregation for the Smart-Grid", en, in *Privacy Enhancing Technologies*, S. Fischer-Hübner and N. Hopper, Eds., Berlin, Heidelberg: Springer, 2011, pp. 175–191, ISBN: 978-3-642-22263-4. DOI: 10.1007/978-3-642-22263-4_10.

[8] A. Rial and G. Danezis, "Privacy-preserving smart metering", in *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, ser. WPES '11, New York, NY, USA: Association for Computing Machinery, Oct. 2011, pp. 49–60, ISBN: 978-1-4503-1002-4. DOI: 10.1145/2046556.2046564. [Online]. Available: https://dl.acm.org/doi/10.1145/2046556.2046564 (visited on 04/10/2025).

[9] F. El Gohary, B. Stikvoort, and C. Bartusch, "Evaluating demand charges as instruments for managing peak-demand", *Renewable and Sustainable Energy Reviews*, vol. 188, p. 113 876, Dec. 2023, ISSN: 1364-0321. DOI: 10.1016/j.rser.2023.113876. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1364032123007347 (visited on 06/04/2025).

[10] J. Swalens, L. Hoste, E. H. Beni, and L. Trappeniers, "zkStream: A Framework for Trustworthy Stream Processing", in *Proceedings of the 25th International Middleware Conference*, ser. Middleware '24, New York, NY, USA: Association for Computing Machinery, Dec. 2024, pp. 252–265, ISBN: 979-8-4007-0623-3. DOI: 10.1145/3652892.3700763. [Online]. Available: https://dl.acm.org/doi/10.1145/3652892.3700763 (visited on 06/03/2025).

[11] X. Salleras and V. Daza, *ZPiE: Zero-knowledge Proofs in Embedded systems*, Publication info: Published elsewhere. Mathematics, 2021. [Online]. Available: https://eprint.iacr.org/2021/1382 (visited on 03/02/2025).

[12] A. Berentsen, J. Lenzi, and R. Nyffenegger, "A Walk-through of a Simple Zk-STARK Proof", en, *SSRN Electronic Journal*, 2022, ISSN: 1556-5068. DOI: 10.2139/ssrn.4308637. [Online]. Available: https://www.ssrn.com/abstract=4308637 (visited on 02/28/2025).

[13] X. Sun, F. R. Yu, P. Zhang, Z. Sun, W. Xie, and X. Peng, "A Survey on Zero-Knowledge Proof in Blockchain", *IEEE Network*, vol. 35, no. 4, pp. 198–205, Jul. 2021, Conference Name: IEEE Network, ISSN: 1558-156X. DOI: 10.1109/MNET.011.2000473. [Online]. Available: https://ieeexplore.ieee.org/document/9520375 (visited on 02/20/2025).

[14] *Proof systems - The halo2 Book*. [Online]. Available: https://zcash.github.io/halo2/concepts/proofs.html?utm_source=chatgpt.com (visited on 05/28/2025).

[15] A. Kothapalli, S. Setty, and I. Tzialla, "Nova: Recursive Zero-Knowledge Arguments from Folding Schemes", en, in *Advances in Cryptology – CRYPTO 2022*, Y. Dodis and T. Shrimpton, Eds., Cham: Springer Nature Switzerland, 2022, pp. 359–388, ISBN: 978-3-031-15985-5. DOI: 10.1007/978-3-031-15985-5_13.

[16] S. Bowe, J. Grigg, and D. Hopwood, *Recursive Proof Composition without a Trusted Setup*, Publication info: Preprint. MINOR revision., 2019. [Online]. Available: https://eprint.iacr.org/2019/1021 (visited on 04/19/2025).

[17] S. Deng and B. Du, *zkTree: A Zero-Knowledge Recursion Tree with ZKP Membership Proofs*, Publication info: Preprint., 2023. [Online]. Available: https://eprint.iacr.org/2023/208 (visited on 04/18/2025).

[18] T. Miyamae, F. Kozakura, M. Nakamura, *et al.*, "ZGridBC: Zero-Knowledge Proof based Scalable and Private Blockchain Platform for Smart Grid", in *2021 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, May 2021, pp. 1–3. DOI: 10.1109/ICBC51069.2021.9461122. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9461122 (visited on 04/06/2025).

[19] J. Eberhardt, M. Peise, D.-H. Kim, and S. Tai, "Privacy-Preserving Netting in Local Energy Grids", in *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, May 2020, pp. 1–9. DOI: 10.1109/ICBC48266.2020.9169440. [Online]. Available: https://ieeexplore.ieee.org/document/9169440 (visited on 04/10/2025).

[20] J. Eberhardt and S. Tai, "ZoKrates - Scalable Privacy-Preserving Off-Chain Computations", in *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, Jul. 2018, pp. 1084–1091. DOI: 10.1109/Cybermatics_2018.2018.00199. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8726497 (visited on 04/10/2025).

[21] *ZKSnarks entmystifizieren: SNARKs für Anfänger erklärt – GBA Global*, de. [Online]. Available: https://gbaglobal.org/de/Blog/2023/07/09/Zksnarks-Snarks-entmystifizieren%2C-erkl%C3%A4rt-f%C3%BCr-Anf%C3%A4nger/ (visited on 06/03/2025).

[22] K. Schiller, *Was ist zk-SNARKs und Zero Knowledge Proof?*, de, Oct. 2018. [Online]. Available: https://blockchainwelt.de/zk-snarks-und-zero-knowledge-proof/ (visited on 06/03/2025).

[23] N. Bitansky, R. Canetti, A. Chiesa, and E. Tromer, *Recursive Composition and Bootstrapping for SNARKs and Proof-Carrying Data*, Publication info: Published elsewhere. Unknown where it was published, 2012. [Online]. Available: https://eprint.iacr.org/2012/095 (visited on 04/19/2025).

[24] S. Samudrala, J. Wu, C. Chen, *et al.*, "Performance Analysis of Zero-Knowledge Proofs", in *2024 IEEE International Symposium on Workload Characterization (IISWC)*, ISSN: 2835-2238, Sep. 2024, pp. 144–155. DOI: 10.1109/IISWC63097.2024.00022. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10763818 (visited on 03/06/2025).

[25] S. Ebrahimi and P. Hassanizadeh, "From Interaction to Independence: zkSNARKs for Transparent and Non-Interactive Remote Attestation", en, in *Proceedings 2024 Network and Distributed System Security Symposium*, San Diego, CA, USA: Internet Society, 2024, ISBN: 978-1-891562-93-8. DOI: 10.14722/ndss.2024.24815. [Online]. Available: https://www.ndss-symposium.org/wp-content/uploads/2024-815-paper.pdf (visited on 05/08/2025).

[26] H. B. Debes, E. Dushku, T. Giannetsos, and A. Marandi, "ZEKRA: Zero-Knowledge Control-Flow Attestation", in *Proceedings of the 2023 ACM Asia Conference on Computer and Communications Security*, ser. ASIA CCS '23, New York, NY, USA: Association for Computing Machinery, Jul. 2023, pp. 357–371, ISBN: 979-8-4007-0098-9. DOI: 10.1145/3579856.3582833. [Online]. Available: https://dl.acm.org/doi/10.1145/3579856.3582833 (visited on 05/08/2025).