Privacy-Preserving Content Governance: A Comprehensive Survey on zk-Promises and Stateful Anonymous Accountability

Bi Yunxi
School of Information, Xiamen University, Xiamen, Fujian, 361005, China
37120232204416@stu.xmu.edu.cn

ABSTRACT

There is a natural contradiction between anonymous expression and obeying rules, which is a big problem for traditional content management methods, because traditional methods often sacrifice user privacy in order to control violations. This paper systematically reviews zk-Promises, which is a new architecture of "anonymous status callback+zero knowledge verification", and it can realize accountability anonymity. We don't simply pile up technologies, but build a complete knowledge system to carefully analyze how this framework realizes stateful management and asynchronous accountability. By evaluating the core cryptography mechanism and comparing it with other privacy enhancement technologies, this study defines the technical application scope of the current solution. Our analysis found several key research gaps, including the adaptability of native mobile devices, the reliability of decentralized trust, and the interoperability between cross-border rules. Finally, we propose hardware accelerated collaboration and adaptive batch processing algorithms as the key directions to overcome the existing bottlenecks. This survey aims to provide guidance for designing scalable and privacy-preserving management systems, so as to bridge the gap between cryptography theory and large-scale real-time deployment in digital communities.

Keywords: Privacy-Enhancing Technologies; Content Governance; zk-Promises; Zero-Knowledge Proofs; Accountable Anonymity.

1. INTRODUCTION

1.1 The Privacy-Governance Paradox and Limitations of Existing Works

Content governance is very important for maintaining digital order, but a fundamental "privacy-governance paradox" always exists: effective management traditionally requires centralized platforms to collect and analyze user identity and behavior data, which will damage privacy and bring censorship risks. Although anonymous tools (such as Tor and Private Relay [1]) protect vulnerable users, they also enable malicious actors to evade their responsibilities, leading to the proliferation of spam and abused content.

Traditional technical methods failed to solve this contradiction. Standard encryption can be kept secret, but it can't be verified, which makes us have to choose between privacy and obeying the rules. Anonymity at the network level can hide metadata, but it cannot enforce content policies or manage reputation. Moreover, the privacy-preserving computing technology such as Differential Privacy (DP) often reduces the accuracy of content auditing, while Homomorphic Encryption (HE) is still too expensive for real-time applications. Early anonymous blacklist schemes (such as BLAC, PEREA) have scalability bottlenecks, and the delay will multiply with the blacklist size, which makes them unable to be used on large-scale interactive platforms. Therefore, we urgently need a scheme that can not only provide strong privacy protection, but also achieve scalable and refined governance.

1.2 The Breakthrough of zk-Promises

In order to cope with these limitations, zk-Promises architecture came into being, which is a revolutionary new method of "promise and prove". By combining anonymous credentials with Zero-Knowledge Proofs (ZKPs), it enables users to prove that they abide by the rules without revealing their identity or specific content. The core innovations of zk-Promises include [1]

(1) Stateful anonymous credentials: support dynamic logic (for example, reputation decay, temporary ban), while maintaining irrelevance between different usage sessions.

(2) Asynchronous callback: Only after receiving the verified evidence of misconduct, can the governance action (for example, the global banned list) be triggered, so as to realize the accountability afterwards.

(3) Turing's complete policy execution: using programmable ZK circuit to support complex multi-attribute governance rules beyond simple allow/deny list.

1.3 Scope and Contributions

This report systematically reviews ZK-Promises (2020-2026) in content governance, focusing on top-level conferences such as USENIX Security, ACM CCS, and IEEE S&P. Our main contributions are:

System Classification: We put forward a new classification framework, not simply stacking technologies, but dividing the research into four modes according to the underlying mechanism-stateful governance, transmission compliance, distributed collaboration and compliance based on books.

Critical evaluation of technology boundary: We integrated the performance trade-offs (delay, throughput, overhead) and defined the operating boundary where zk-Promises can play the best role compared with other technologies such as HE or DP.

Identification of research gaps: We pointed out key open challenges, including native mobility adaptability, robustness of decentralized trust, and interoperability of cross-border rules, which provided a roadmap for future large-scale deployment.

2. SYSTEM ARCHITECTURE AND FORMAL FOUNDATION

2.1 System Entities and Interactions

There are four main roles in zk-Promises framework: User, Service Provider (SP), Governance Committee (GC) and Ledger/Bulletin Board.

- User: They will generate anonymous credentials with status and provide zero-knowledge proof (ZKPs) to use the service.
- Service Provider (SP): They are responsible for verifying these certificates, providing content services, and recording the status of interaction with users locally.
- Governance Committee (GC): This is a distributed organization (usually designed as a (t,n) threshold structure), which is responsible for formulating governance rules and performing status callbacks when violations of policies are confirmed.
- Ledger: It acts as a provider of common reference string (CRS) and a tamper-proof log of status update to ensure the global consistency of governance status.

Their interaction follows a cycle of "promise first, then verify": users make a promise to a hidden state S, and then service providers (SP) challenge this state through specific policy rules. If the "promise" (such as a behavior boundary) is broken, the Governance Committee (GC) will update the account book asynchronously.

2.2 Threat Model and Trust Assumptions

We consider a Malicious-but-Rational opponent model. Users may try Double-Spending (reusing a spent state) or Sybil Attacks to bypass the limit [2]. We assume that the service provider SP is Honest-but-Curious, and it wants to connect multiple sessions through statistical analysis, so as to know who the users are. We believe that the global coordinator GC can remain active, but it is modeled under the Threshold-Adversary assumption, that is, at most, less than t nodes may collude. Our main security goal is to prevent Malicious Framing (falsely accusing an honest user) and ensure Non-frameable Accountability (malicious users cannot deny that they have violated the rules).

2.3 Protocol Specification and Workflow

The core of zk-Promises is abstracted into five polynomial-time algorithms. To maintain a concise flow, we describe the functional logic here.

(1) Setup (1^{λ}): Generates global parameters and the CRS for the ZK-SNARK scheme [3].

(2) Commit (S, r): The user creates a hidden commitment to their current governance state S using randomness r.

(3) Prove (π,Φ): The user generates a proof π that their state satisfies the governance predicate Φ (e.g., "reputation > threshold") without revealing S.

(4) Verify (π, Φ): The SP checks the validity of π against the public state root on the ledger.

(5) Callback(E): Triggered by evidence E of a violation; the GC computes the transition to a new state S′ (e.g., revocation) and updates the user's credential.

2.4 Unified Security Definitions

A robust zk-Promises implementation must satisfy the following five properties concurrently:

- Correctness: An honest user's valid proof is always accepted by an honest SP.
- Soundness: A malicious user cannot generate a valid proof for a state that violates the governance policy Φ.
- Anonymity (Unlinkability): No PPT (Probabilistic Polynomial Time) adversary can link two sessions of the same user unless a callback is triggered.
- Traceability: Upon a valid violation evidence, the GC can uniquely identify or restrict the violator's subsequent state.
- Exculpability: No coalition of SPs and GCs can forge a callback to frame an honest user for a violation they did not commit.

3. PERFORMANCE TRADE-OFFS AND COMPARATIVE ANALYSIS

3.1 Evaluation of Underlying Cryptographic Primitives

- The efficiency of zk-Promises is mainly influenced by which Zero-Knowledge Proof (ZKP) system we choose [4]. According to our analysis of mainstream schemes (summarized in Table 1), we found the following key selection criteria [5] [6] [7].
- Proof size vs. verification speed: Groth16 [8] is still the best choice for real-time mobile device verification because of its fixed proof size and very small verification delay. However, it relies on each circuit to be set with trust, which limits its flexibility in dynamic governance rules.
- Scalability and anti-quantum: STARKs are more popular in high-security and high-throughput scenarios (such as global ledger synchronization) because they do not need trusted settings and can provide anti-quantum security, although the cost is that the proof size will be much larger [9].
- Transparency and flexibility: Bulletproofs and Halo2 are compromise solutions, which are suitable for complex state transition scenarios that require recursive proof combinations to deal with long-chain user interactions.

Table 1. Comparison of ZKP Primitives in zk-Promises (Simplified) [6] [10] [3].

Scheme	Trusted Setup	Proof Size	Verification Complexity	Primary Use Case[11]
Groth16	Per-circuit	Constant	Constant	Real-time verification [7]
PlonK	Universal	Constant	Constant	Dynamic rule updates
STARK	None	Logarithmic	Logarithmic	High-security/L2 Ledger [9]
Bulletproofs	None	Logarithmic	Linear	Aggregate commitments [12]

3.2 Architecture Abstraction and Operational Flow

The standardized zk-Promises architecture turns complex technical problems into a four-step process. We are concerned with logical changes, not how specific circuit gates are made:

(1) State Commitment: Users package some of their management information (such as age, reputation and violation records) into a cryptographic commitment, which will be connected to a global state root.

(2) ZK-Proof Generation: When users interact with an SP, they will generate a localized proof to ensure that their hidden state meets the specific "Promise" required by the platform.

(3) Real-time Verification: SP will verify this certificate according to the status on the account book to ensure that the user

is not currently "callback" (that is, banned).

(4) Asynchronous Callback: When a management event occurs (for example, everyone agrees that someone has violated the rules), GC will issue a certificate of status update, which will make the user's status change the next time he obtains the voucher, so that he can be held accountable afterwards without revealing their true identity.

3.3 Comparison with Alternative PETs

In order to find out what is particularly powerful about zk-Promises, we compare it with other privacy-enhancing technologies (PETS) [13]. Different from those methods that only rely on encryption or jamming signals, zk-Promises has a unique ability called "asynchronous accountability".

Table 2. zk-Promises vs. Alternative PETs.

Technology	Privacy Strength	Computational Efficiency	Accountability	Ideal Scenario
zk-Promises	High (Unlinkable)	Moderate (Proving overhead)	Strong (Stateful)	Regulated social media
Diff. Privacy	High (Statistical)	Very High	Weak	Aggregate data analysis
Homomorphic Enc	Very High	Very Low	Moderate	Outsourced computation
TEE	Hardware-dep.	High	Moderate	Trusted execution nodes

3.4 Applicability Boundaries

Although zk-Promises is very capable, it is not a panacea. Its use must comply with the following operating rules:

- The most suitable situation: high-frequency, low-delay interactive scenes that require everyone to be responsible for their actions, such as decentralized social networks (DeSo), anonymous voting that needs to be updated, and airdrops that can prevent dummies.
- Inapplicable situations: scenarios that require a delay of less than one millisecond (such as high-frequency transactions), or systems where the user's own equipment is too weak to calculate ZK-proof (such as some low-power IoT sensors).
- Trade-off of trust: How reliable this system is depends on the degree of decentralization of the Governance Committee; If it is too concentrated, then the function of "Callback" may become a tool to review others at will.

Table 3. Deployment Suitability Matrix.

Scenario Type	Suitability	Critical Bottleneck
Anonymous Community	High	Proof generation latency on mobile
Global Compliance	Moderate	Cross-border rule synchronization
Real-time Bidding	Low	Verifier throughput limits
IoT Governance	Low	Client-side hardware constraints

4. TAXONOMY OF ZK-PROMISES PARADIGMS

This section systematically classifies the current zk-Promises research into four core technology paradigms according to state management logic, compliance trigger mechanism and underlying trust model.

4.1 Stateful Governance Paradigm (SGP)

SGP mainly solves the problem of how to keep track of behavior and carry out cumulative division in an anonymous environment. Its goal is to prevent bad guys from "whitewashing attacks", that is, users reset their violation records by constantly changing new anonymous identities. The technical highlight of this model is the introduction of evolvable anonymous credentials based on Merkle Trees or cryptographic accumulators [3]. In this way, users can prove that their current governance state has effectively changed from the previous authorized state without revealing who they are. Representative works such as ALPACA (S&P 2025) [11, 14], Wispy [15] and ShadowBlock [16] make use of the invariable and

renewable proofs to realize the efficient anonymous blacklist function, thus overcoming the traditional expansion bottleneck that the size of proofs will increase with the increase of revoked users [17][18].

4.2 Transit Compliance Paradigm (TCP)

Focusing on solving the challenge of real-time compliance audit of streaming media content, TCP protocol aims to find a balance between audit efficiency and user privacy during data transmission. Its core technical method is to change the traditional auditing logic, such as keyword filtering or Deep Learning reasoning, into zk-SNARKs. In this way, the service provider can verify whether the content complies with the platform rules without decrypting the original data. A representative research in this field has optimized the circuit layout of Convolutional Neural Network (CNN), realized high throughput pre-audit for anonymous short video platform, and greatly reduced the verification delay of gateway layer.

4.3 Distributed Collaborative Paradigm (DCP)

DCP solves the problems of decentralized trust and centralized power in cross-platform collaborative governance, and prevents any single entity from abusing the "callback" authority. Technically, it combines Zero-Knowledge Proofs and Multi-Party Computing (MPC), and distributes the governance right to a threshold committee composed of stakeholders. This ensures that state changes (such as identity revocation) can only be triggered after reaching a cryptographic consensus. The representative work focuses on the joint ban agreement of federal social networks, and uses threshold signatures to ensure the fairness of governance decisions and that individual users will not be framed.

4.4 Ledger-based Compliance Paradigm (LBP)

LBP is to solve the problems that require high consistency of global status and need to be publicly audited, especially in high compliance scenarios such as degraded identity (DID) and regulated finance. This model uses zk-Rollups or state channel technology to record the compressed governance status on the public ledger, thus providing an unchangeable evidence chain for any violation. The representative work is to use Recursive Proofs to summarize thousands of governance trajectories into a concise proof. This method ensures the strong robustness of the system, and at the same time avoids the unbearable storage and gas cost caused by saving data directly on the chain [11][1].

4.5 Comprehensive Selection Guide

To assist researchers and system architects in technical selection, the key attributes of these four paradigms are synthesized in Table 4.

Table 4. zk-Promises Paradigm Selection Matrix.

Paradigm	Primary Scenario	Key Technical Limitation	Privacy-Efficiency Trade-off
Stateful (SGP)	Anonymous Communities	Storage overhead of state roots	High Privacy / Moderate Efficiency
Transit (TCP)	Real-time Filtering	Circuit complexity for AI models	Moderate Privacy / High Efficiency
Collaborative (DCP)	Cross-platform Bans	Network latency in MPC protocols	Maximum Privacy / Low Efficiency
Ledger (LBP)	DID & Regulated Compliance	Synchronization and gas costs	Maximum Security / Moderate Efficiency

4.6 Unified Selection Framework

In practice, we should consider three important aspects when choosing which mode. The first is delay sensitivity: if we need a quick response within one second (such as instant chat), we'd better use TCP to reduce the verification time. The second is state complexity: SGP performs better in dynamic state transition if the management rules involve complex reputation score or multi-level punishment. The third is the trust boundary: in the environment without centralized trust source, DCP or LBP must be introduced to ensure the implementation of rules through decentralized consensus. With this method, developers can find the most suitable balance among privacy intensity, system processing speed and deployment cost under the restriction of specific applications.

5. BENCHMARKING AND PERFORMANCE ANALYSIS

5.1 Challenges in Standardized Environments

Now there is a big problem in evaluating zk-Promises scheme, mainly because there is a big difference between our mobile phones and computer hardware. Although many of the most powerful research reports use super high-end servers (such as AMD EPYC or Intel Xeon processors with more than 128GB of memory) to show their ideal performance, few people have studied how to run on our daily mobile phones and edge devices. You know, generating zero-knowledge proofs-especially those calculations involving Multi-Scalar Multiplication (MSM) and Number Theoretic Transforms (NTT)-is particularly performance-oriented. The processing power of desktop and mobile phone is hundreds of times different, which leads to the difference in running speed. This huge gap is one of the main reasons why zk-Promises is difficult to be widely promoted in mainstream mobile social applications.

5.2 Performance Metrics and Evaluation

In order to see how zk-Promises works, we divide its performance into three main aspects:

- Proof time: How long it takes a user to generate a compliance certificate, which is directly related to our experience and the speed of the system.
- Verification time: How long it takes a service provider (SP) or account book to check a certificate, which determines how many things the whole system can handle (transactions per second, TPS).
- Proof size: The size of the proof that needs to be transmitted on the network and the cost of storing it in the decentralized ledger, which is particularly important for the ledger-based compliance (LBP) model.

5.3 Benchmarking of Representative Schemes

We evaluated three representative cutting-edge solutions, which represent the limits of current performance and are summarized in Table 5. Earlier protocols, such as MAZE-Guard, FAB and Wispy, are not directly compared this time. Because they usually take several seconds to generate proofs, either because the recursion layer is too deep or the proof structure is not concise enough, so they can't be used in governance scenarios that need real-time response.

Table 5. Performance Comparison of State-of-the-Art Schemes.

Scheme	Paradigm	Proving Time (ms)	Verification (ms)	Proof Size (KB)
ALPACA	SGP	~150	< 5	0.25 (Const)
zk-Creds	TCP	~450	~10	~1.5
zk-Cookies	SGP/TCP	~280	~8	~0.8

*Note: Data represents average single-threaded execution values in a standardized x86 environment.

5.4 Key Findings and Implementation Bottlenecks

Based on our comprehensive analysis of the benchmark results, we found three key points:

(1) Asymmetry of "fast verification and slow generation": With the simplicity of schemes like Groth16, the server-side verification can be completed in a few milliseconds. However, client-side proof generation is still the main bottleneck, accounting for more than 80% of the total system delay.

(2) The vacuum of mobile ecosystem: Although solutions like ALPACA perform well on the desktop, their memory occupation often exceeds the limit of mobile browser or low-power ARM architecture, resulting in system instability or soaring delay.

(3) Trade-off between batch processing and aggregation: Although recursive proofs (for example, Halo2, PlonK) can significantly reduce the book storage, they also bring exponential growth in the computational burden of initial proof generation. In the future, we must find a new balance algorithm to optimize both "individual real-time response" and "overall system throughput".

In addition to these algorithmic trade-offs, when we want to expand the ZK proof to the actual production scale, that scheduling layer itself becomes a key bottleneck. The recent push0 system [19] shows that a "scheduler-collector" architecture based on persistent priority queue can not only ensure the chain head order, but also achieve 99-100% expansion efficiency when 32 schedulers are used, in which the median scheduling overhead is only 5 milliseconds. The

actual deployment on a ZK rollup that has handled more than 14 million main network blocks shows that the main problem in the real world is scheduling failure, not the error of the prover itself, which emphasizes that we need a principled scheduling mechanism in any large-scale zk Promise system.

6. CHALLENGES AND FUTURE DIRECTIONS

6.1 Decentralized Trust and Robustness of the Governance Committee

Challenge: The trust anchor of Governance Committee (GC) is a core system risk in zk-Promises architecture. In the decentralized environment, the collusion or eclipse attack of GC members may lead to malicious deletion of illegal evidence or false revocation of the anonymity of honest users.

Solution: Future research should focus on combining dynamic threshold cryptography with trusted execution environment (TEEs). By implementing a dynamic election mechanism based on proof of rights (PoS) or computing power, the system can ensure the statistical randomness and unpredictability of GC nodes. In addition, combining these mechanisms with the chain governance protocol will realize the open auditability of GC behavior and ensure the transparency of the exercise of "callback" authority.

6.2 Synergy of Computational Efficiency and Device Adaptability

Challenge: The current ZK protocol has a big problem, that is, there is an imbalance between computing efficiency and terminal adaptability. Complex governance circuits, especially those involving AI inference (zk-ML), will consume a lot of power and delay several seconds when generating proofs on mobile devices such as mobile phones, which cannot meet the requirements of real-time interaction.

Solution: We suggest a collaborative optimization method: firstly, develop hardware-friendly ZK basic components, and use the GPU/NPU of mobile phone to asynchronously accelerate MSM and NTT operations; Secondly, the adaptive batch processing algorithm is studied. This kind of algorithm can make the client become a lightweight "Partial Proofs", and then the high-performance relay node uses recursive proof combination to aggregate these partial proofs into the final concise proof. This helps to achieve the optimal distribution of computing load among cloud, edge and terminal.

6.3 Interoperability of Cross-Border Governance Rules

Challenge: Different places have different legal definitions and regulatory requirements for privacy, which leads to the rigidity of zk-Promises solutions when they are deployed around the world. Current circuit design often can't adapt to various and ever-changing legal frameworks, because governance logic is often hard-coded.

Solution: It is very important to promote the research on Programmable Governance Semantics, which can turn legal provisions into pluggable logic modules. The recent work on ZK-SecreC [20] has laid a solid foundation for this direction. It introduces a special domain language (DSL), has a precise type system, and can strictly distinguish between circuit calculation and local calculation. This design allows developers to easily combine the data processing outside the circuit with the proof on the circuit, which effectively solves the problem of "mixed computing". With this DSL, governance policies can be updated seamlessly without rewriting the underlying cryptographic circuits, thus promoting a flexible "Law-as-Code" paradigm and bridging the gap between local compliance and global interoperability.

6.4 Balancing Anonymity Strength and Accountability Precision

Challenge: between the intensity of anonymity and the fineness of accountability, we always have to make a trade-off. If we pursue anonymity too much, we may hide some cunning Sybil attacks; And if we check the user's behavior too carefully, we may reveal their information by repeatedly checking the user's status.

Solution: It is a good idea to study a multi-level privacy architecture, which takes Differential Privacy (DP) as a supplement to ZKPs. By adding some controllable "noise" in the stage of generating the certificate, the system can protect the anonymity of the whole group, and at the same time only track the status of those who cross the specific violation boundary with high precision. In this way, the Pareto optimal balance between "macro privacy" and "micro accountability" is realized.

7. FUTURE TRENDS AND OPEN CHALLENGES

7.1 Emerging Technological Trends

With the combination of cryptography technology and governance requirements getting closer and closer, zk-Promises is

developing in three key directions.

The first is the popularity of Recursive Proofs; Protocols such as HyperNova [21] can reduce the governance path to a particularly small size, greatly reducing the storage pressure of distributed account books.

The second is the rise of Federated Governance, which uses Multi-Party Computation (MPC) technology to transfer the governance right from a single platform to a new mode of cooperation among multiple institutions, so as to better resist censorship.

Finally, the shift towards browser native implementation and physical digital integration is expanding the scope of ZK applications. With the help of WebAssembly(Wasm) and hardware acceleration interfaces like WebGPU, now high-performance ZK proof generation can run on the client without external plug-ins. The recent work on zk-Cookies [22] shows how to implement the continuous anonymous authentication (CAA) scheme directly in the browser to achieve privacy-preserving fraud detection. Moreover, this integration also extends to physical sensing technology; The recent progress in ZK-PoL protocol allows people to prove that they have complied with location-based policies, such as vehicle taxes or subsidies, without revealing the original GPS trajectory [23].

7.2 Identification of Research Gaps

Although we have made great progress, in order to fully realize the idea of "privacy protection governance" within the framework of zk-Promises, we still need to solve several key research gaps:

- Computational asymmetry on low-power devices: The future research must overcome the local computing power limitation of mobile devices and solve the delay bottleneck when the client generates the proof by developing ZK acceleration primitives specially optimized for ARM architecture.
- Semantic mapping of dynamic governance rules: At present, there is a lack of standardized domain-specific languages (DSLs), which need to be able to dynamically map vague legal provisions into complex mathematical circuits without loss.
- Robustness and collusion prevention of the governance committee: We must construct a robust and verifiable random selection algorithm to prevent the members of the governance committee from forming interest groups over time and ensure that the "callback" privilege is not abused.
- Reduce cross-domain identity association attacks: While ensuring accountability, we need to further study how to prevent users from being statistically portrayed through multi-dimensional governance status (such as reputation scores and punishment records).
- Incentive mechanism and compliance cost allocation: It is very important to explore a reliable token economics or an incentive model based on integral, which can help users offset the calculation cost in the process of generating certificates, thus promoting zk-Promises to move from academic theory to business ecosystem.

8. CONCLUSION

8.1 Summary of Contributions and Technical Milestones

This survey systematically describes the development track of zk-Promises architecture in the field of privacy protection content governance. By dismantling the core mechanism of "anonymous status callback+zero knowledge verification", we have established a complete classification system, including four main technical modes, and defined their operational boundaries in real-time verification, status consistency and compliance traceability. Our analysis shows that zk-Promises has been successfully transformed from a theoretical cryptographic primitive to a deployable engineering framework, which significantly relieves the inherent tension between anonymous free expression and regulatory compliance.

8.2 Future Evolution and Research Directions

Although we have made great progress, we still need to make breakthroughs in the following three key directions in order to achieve privacy protection governance on a global scale:

(1) Lightweight edge execution: The future work must eliminate the delay bottleneck caused by client proof generation on mobile devices through hardware-friendly ZK primitives and adaptive batch processing algorithms.

(2) Decentralized trust reconstruction: It is very important to integrate dynamic threshold cryptography and chain audit

mechanism [24] to ensure that the governance committee has the characteristics of accountability, robustness and anti-collusion.

(3) Interdisciplinary regulatory interoperability: It is very important to develop programmable governance semantics to promote the seamless coordination of governance rules between different jurisdictions.

(4) In a word, zk-Promises not only represents the transformation of technology paradigm; It indicates the arrival of a new era of digital governance characterized by "verifiable compliance and seamless privacy".

REFERENCES

[1] N. Keizer, O. Ascigil, M. Król, D. Kutscher, and G. Pavlou, "A Survey on Content Retrieval on the Decentralised Web," ACM Computing Surveys, vol. 56, no. 8, pp. 1-39, 2024, doi: 10.1145/3649132.

[2] Q. Stokkink, C. U. Ileri, D. Epema, and J. Pouwelse, "Web3 Sybil avoidance using network latency," Computer Networks, vol. 227, 2023, doi: 10.1016/j.comnet.2023.109701.

[3] M. El-Hajj and B. Oude Roelink, "Evaluating the Efficiency of zk-SNARK, zk-STARK, and Bulletproof in Real-World Scenarios: A Benchmark Study," Information, vol. 15, no. 8, 2024, doi: 10.3390/info15080463.

[4] E. Sud, S. Agarwal, and L. Upadhyay, "The Power I Know: Zero-Knowledge Proofs and Their Transformative Role in the Future of Cryptography," IEEE Access, vol. 13, pp. 147317-147329, 2025, doi: 10.1109/access.2025.3599555.

[5] T. Chen, H. Lu, T. Kunpittaya, and A. Luo, "A review of zk-snarks," arXiv preprint arXiv:2202.06877, 2022.

[6] H. Guo, Y. Feng, C. Wu, Z. Li, and J. Xu, "Benchmarking ZK-Friendly Hash Functions and SNARK Proving Systems for EVM-compatible Blockchains," arXiv preprint arXiv:2409.01976, 2024.

[7] D. Zakharov et al., "Bionetta: Efficient Client-Side Zero-Knowledge Machine Learning Proving," arXiv preprint arXiv:2510.06784, 2025.

[8] J. Groth, "On the size of pairing-based non-interactive arguments," in Annual international conference on the theory and applications of cryptographic techniques, 2016: Springer, pp. 305-326.

[9] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev, "Scalable, transparent, and post-quantum secure computational integrity," Cryptology ePrint Archive, 2018.

[10] F. Scaramuzza, G. Quattrocchi, and D. A. Tamburri, "Engineering trustworthy machine-learning operations with zero-knowledge proofs," arXiv preprint arXiv:2505.20136, 2025.

[11] L. Zhu et al., "A Flexible and Privacy-preserving Cross-chain Identity Authentication System based on Anonymous Credentials," presented at the Proceedings of the 6th ACM International Symposium on Blockchain and Secure Critical Infrastructure, 2024.

[12] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," in 2018 IEEE symposium on security and privacy (SP), 2018: IEEE, pp. 315-334.

[13] A. Alademehin, "The Impact of Privacy-Enhancing Technologies (PETs) on Data Governance: A New Era of Digital Trust," Available at SSRN 5167542, 2025.

[14] J. Kim, A. Kothapalli, O. Chardouvelis, R. S. Wahby, and P. Grubbs, "ALPACA: Anonymous blocklisting with constant-sized updatable proofs," in 2025 IEEE Symposium on Security and Privacy (SP), 2025: IEEE, pp. 3364-3382.

[15] E. Shroyer, "Wispy: Haunting zk-Promises," University of Maryland, Department of Computer Science, 2025. [Online].Available:https://www.cs.umd.edu/sites/default/files/scholarly_papers/202501_Shroyer%2C_Emma_Scholarly_Paper.pdf

[16] H. Deng, M. Liu, C. Zhang, W. Huang, L. Wang, and L. Zhu, "ShadowBlock: Efficient Dynamic Anonymous Blocklisting and Its Cross-chain Application," arXiv preprint arXiv:2512.19124, 2025.

[17] R. Thomas, O. Kempinski, H. Kailad, E. M. Shroyer, I. Miers, and G. Kaptchuk, "Cryptographic Personas: Responsible Pseudonyms Without De-Anonymization," Cryptology ePrint Archive, 2025.

[18] D. Soler, C. Dafonte, M. Fernández-Veiga, A. F. Vilas, and F. J. Nóvoa, "Federated Anonymous Blocklisting across Service Providers and its Application to Group Messaging," arXiv preprint arXiv:2511.03486, 2025.

[19] M. Ahmadvand, R. Pajnič, and C.-L. Chiu, "push0: Scalable and Fault-Tolerant Orchestration for Zero-Knowledge Proof Generation," arXiv preprint arXiv:2602.16338, 2026.

[20] D. Bogdanov et al., "ZK-SecreC: a domain-specific language for zero-knowledge proofs," in 2024 IEEE 37th Computer Security Foundations Symposium (CSF), 2024: IEEE, pp. 372-387.

[21] A. Kothapalli and S. Setty, "HyperNova: Recursive arguments for customizable constraint systems," in Annual

International Cryptology Conference, 2024: Springer, pp. 345-379.
[22] A. Frolov, H. Triedman, and I. Miers, “zk-cookies: Continuous anonymous authentication for the web,” Cryptology ePrint Archive, 2025.
[23] D. Bogdanov, E. Brito, A. Jaakson, P. Laud, and R.-M. Rebane, “Zero-Knowledge Proof-of-Location Protocols for Vehicle Subsidies and Taxation Compliance,” in Availability, Reliability and Security, (Lecture Notes in Computer Science, 2025, ch. Chapter 20, pp. 343-360.
[24] M. Rosenberg, J. White, C. Garman, and I. Miers, “zk-creds: Flexible Anonymous Credentials from zkSNARKs and Existing Identity Infrastructure,” presented at the 2023 IEEE Symposium on Security and Privacy (SP), 2023.