https://doi.org/10.58245/ipsi.tir.2603.04

Forward-Secure Blockchain Architecture for National-Scale Electronic Voting: Enhancing Institutional Sustainability and Democratic Resilience

Stanojević, Nebojša; Vukmirović, Dragan; Čomić, Tijana; Stanojević, Dejan; and Ćetković, Miloš

Abstract: National electronic voting systems require robust auditability, non-repudiation, forward secrecy, and fault tolerance under predictable but highly variable workloads. Existing protocols within permissioned blockchain networks usually favor low latency or long-lasting identities of business entities, which are assumptions that do not fully correspond to adversarial election environments.

This paper presents a feasibility study of a time-slot-based permissioned blockchain architecture designed for state elections involving semi-trusted but mutually adversarial validators. The basic mechanism of the system is a block authentication scheme with advanced security, based on ephemeral key pairs: validators publish the public key for block N+1 within block N, after which they use the corresponding private key once to sign the next block and immediately delete it permanently.

As part of the research, a multi-node prototype was implemented as a customized infrastructure of permissioned blockchain in the Node.js environment, with the application of Ed25519 digital signatures, explicit block finalization, peer-to-peer communication between validators and full signature verification by all participants. Empirical evaluation in a simulated wide-area network (WAN) environment shows a stable throughput of approximately 274 transactions per second, with a median finalization (p50) of 10.7 seconds and a 99th percentile (p99) of under 21 seconds.

A projection on an election scenario involving six million voters during the 13:00 voting period indicates an average safety factor load of approximately 2.1. The presented results of the proposed model should be interpreted as a proof of concept obtained in a controlled prototype environment, and not as a complete validation of the production capacity on a geographically distributed election infrastructure. The findings indicate technical feasibility under defined assumptions and demonstrate how domain-specific transaction clustering, ephemeral key authentication, and explicit validator signatures can support an auditable consensus layer for electronic voting. Out of scope for this current study are client-side privacy protection, coercion resistance, and full end-to-end ballot secrecy. Our structural architecture explicitly hardens the consensus and ledger layers instead.

Index Terms: electronic voting; permissioned blockchain; forward security; ephemeral keys; auditability consensus protocols; blockchain infrastructure; scalability evaluation

1. INTRODUCTION

Electronic voting promises to make public elections both user-friendly and straightforward to audit, yet actual national-scale deployments remain rare. The fundamental hurdle is temporal: an election system must operate flawlessly within a narrow, legally mandated voting window, while its technical records must remain irrefutably convincing long after the polls close. Because legal disputes often surface weeks or months after the fact, the underlying data logs must remain fully verifiable over time. While blockchain technology is frequently proposed to secure these records through append-only structures and distributed validation, standard implementations fail to safeguard the system against long-term vulnerabilities.

To build this unalterable archive, researchers frequently turn to blockchain infrastructure, drawn by its decentralized validation and append-only architecture. However, standard blockchain design is not always suitable for elections. Public blockchains solve different problems because they rely on open participation and economic incentives. On the other hand, private blockchains (with access permissions) are closer to the needs of public institutions. However, they usually rely on well-known validators, whose signing keys are valid for a longer period of time. It is not an ideal solution. Validators can be politically opposed and are also attractive targets for attacks.

This paper addresses a specific problem: the subsequent compromise of validator keys. What if the same key is used for multiple blocks or over a long period? An attacker who gets to it after the election can create a fake history. At the very least, it may cast doubt on the authenticity of the electoral register. In business systems, this is handled by contracts and internal procedures. However, with elections, the consequences for society and politics are incomparably greater.

That is why we study small and controlled consensus layer design. In it, the block signing key lasts for a very short time. Our idea is not to build a complete e-voting system. Voter authentication, secrecy, coercion resistance, and practicality are left to the application layer. Our goal is narrower. We want to examine whether a permissioned blockchain can use simple block authentication with forward secrecy while maintaining a realistic data processing speed at the national level.

The paper has four contributions.

First, it proposes a one-block-ahead ephemeral-key mechanism for block authentication.

Second, it uses a slot-based workflow that fits the bounded and bursty nature of voting.

Third, the architecture retains explicit validator signatures inside the blocks, allowing independent observers to audit the entire chain long after election day.

Fourth, our experimental evaluation transparently outlines the scope of the prototype, distinguishing the fully coded components from the parameters that were simulated.

2. Related Work

2.1. Permissioned Blockchain Consensus

BFT consensus protocols like PBFT, Tendermint, and HotStuff serve as standard foundations for permissioned ledgers, providing deterministic finality under bounded Byzantine failures [5], [6], [17]. Advanced architectures like Narwhal and Bullshark further boost throughput by decoupled processing of data dissemination and ordering [8]. While highly effective, these frameworks fundamentally assume that validator keys remain secure and unaltered under the broader system governance.

That perspective holds true across many enterprise setups where network nodes are run by corporations bound by legal contracts, making key rotation a routine operational task. When applied to public elections, however, this assumption becomes highly problematic. Even if validators represent established public institutions or political entities, they rarely enjoy universal trust from every voting faction. Consequently, if a long-term signing key is compromised, it triggers a widespread crisis of public legitimacy rather than a simple technical failure.

Hyperledger Fabric serves as a prominent enterprise permissioned blockchain framework [2], utilizing an adaptable architecture that uncouples endorsement, ordering, and validation tasks. It is useful in business. However, this also means that public review depends on complex access and configuration rules. In our design, audit proofs are intentionally closer to the block itself. Each block contains explicit validator signatures. It also contains the information necessary to check the transition of the ephemeral (short-lived) key.

2.2. Blockchain-Based Electronic Voting

Electronic voting is not a new phenomenon; it has been the subject of research for many years. The literature in this area generally indicates that the accuracy of election results is not the only requirement for a reliable electoral system. Equally important are voter privacy, citizen trust, and the ability to subsequently verify the results and the manner in which the electoral process is conducted [4], [7]. One example is Helios, a web-voting system that allows for open audit trails [1]. More recent protocols, such as VERICONDOR, also improve end-to-end verifiability for certain electoral rules [9].

Several authors warn that blockchain is not a complete solution for voting [13]. It makes records resistant to unauthorized changes. However, blockchain alone does not protect the voter's device. It does not prevent coercion, does not hide the vote and does not decide who has the right to vote. We agree with that view. Our work does not solve all voting problems. We focus exclusively on consensus records. The main question is how that record remains reliable if the validator keys are subsequently compromised.

2.3. Forward Security and Key Compromise

Forward-secure signatures have been proposed to reduce the damage caused by key compromise [3], [10]. Fundamentally, exposing the current key material must not grant an attacker the ability to forge valid signatures for historical epochs. While certain formal cryptographic frameworks offer robust security guarantees, they often bring substantial operational complexity to the actual deployment.

Alternatively, simpler strategies achieve practical forward security by restricting key lifetimes and strictly regulating their exposure windows [11]. Our design follows this engineering-oriented path. Rather than introducing a full key-evolving signature scheme, each block is assigned a short-lived signing key whose public component is stored one block in advance.

The transition to post-quantum cryptography makes this topic even more relevant. Measurements of post-quantum signatures in blockchain environments show that signature size and verification costs can affect throughput, propagation, and storage [15]. Our prototype is not post-quantum secure, since it uses Ed25519. The same architecture, however, can be used to study where post-quantum signatures would create additional load in an electoral register.

2.4. Research Gap

The existing literature already discusses developed BFT consensus protocols, verifiable voting protocols, and cryptographic mechanisms that mitigate the consequences of subsequent key discovery. However, it is evident that a permissioned electoral registry model specifically adapted to this environment is missing. Such a system would need to combine predictable block finality, clear evidence for subsequent revision, and limited exposure of validator keys at the block level. This work seeks to fill this gap through the development and testing of a prototype. System and threat model

3. System and Threat Model

3.1. System Model

The proposed system is conceived as a permissioned distributed ledger, designed specifically for the loads characteristic of national elections [13], [14], [16]. The system architecture includes three types of entities: validators, clients and observers. Validators form a fixed set of consensus nodes managed by mutually distrustful institutional actors. These roles may be filled by the electoral commission, political parties, accredited auditors, or civil society groups. Clients submit voting transactions directly, while observers function as read-only nodes that replicate the ledger to verify cryptographic signatures and key transitions.

Our model operates under a partially synchronous network assumption. This means that while individual messages may face temporary transmission delays, their ultimate delivery remains guaranteed within an unspecified, finite time bound. The system assumes a standard BFT threshold at which security is maintained while the number of Byzantine validators satisfies n >= 3f + 1 [6], [12], [17]. The long-term identity keys of the validators are known via the initial public key infrastructure (PKI) but are not used for conventional block signing. Consensus operations rely on short-lived (ephemeral) keys.

3.2. Threat Model and Scope

The threat model assumes that an adversary can control up to f validators, distribute messages within defined partial synchronicity limits, cause individual nodes to crash, or compromise the validator's internal state at time t. The main security goal of the system is to preserve the integrity of the ledger: after a certain block is finalized, later compromise of the validator should not enable retroactive falsification of blocks that were finalized before the very moment of compromise [3], [11].

This paper primarily focuses on the consensus and authentication layer of the ledger. The proposed solution does not include voter anonymity, resistance to coercion, the operation of malicious software (malware) on the client side, the usability of the voting system, or legal identity verification. The above issues are considered orthogonal and must be addressed within the wider application architecture of e-voting, for example, by implementing blind signatures, mix-nets, zero-knowledge proofs or other verifiable voting mechanisms [1], [4], [7], [9].

The temporal execution of these phases is depicted in Figure 2, which emphasizes that the private key eskt+1 remains in the 'Exposure Window' only for the duration of a single slot before the Secure Erase operation renders it permanently unrecoverable [10].

4. Architecture Design

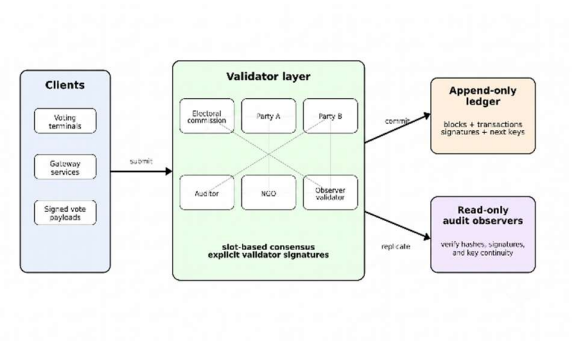

Figure 1. Permissioned blockchain infrastructure for national-scale electronic voting

4.1. Blockchain Infrastructure

The revised implementation is explicitly defined as a custom permissioned blockchain prototype, not as a solution implemented on Hyperledger Fabric [2], Tendermint [5], or any other existing framework. The reason for this design approach lies in the need for experimental control: the developed prototype isolates the proposed authentication mechanism with ephemeral keys and allows the block structure, validator signatures, as well as the life cycle of the keys, to be directly observable and amenable to analysis.

The infrastructure of the subject blockchain consists of an append-only block log, a deterministic slot scheduler, a peer-to-peer communication layer between validators, a transaction pool (mempool), block header verification logic, aggregation of multiple validator signatures, and an ephemeral key cache. Each block structurally contains the height of the block, the hash of the previous block, the group of transactions, the identity of the proposer, the public ephemeral keys for the next block and the explicit signatures of the validators. Validators commit a block only after successfully verifying the previous hash, transaction format, quorum signature and continuity of public ephemeral keys.

4.2. Slot-Based Consensus Model

The time frame of the system is divided into discrete slots S_t. Within each slot, the deterministic leader selection rule assigns exactly one validator with the task of proposing a block B_t. Such a design based on time slots is adequate for application in electronic voting systems, since election traffic is by its nature variable, but legally and time-limited. The primary goal of the proposed solution is not to achieve the maximum theoretical throughput in all operational conditions, but to ensure predictable finalization and stable audit evidence during a defined voting window. Machines within a low-latency institutional network.

4.3. Forward-Structure Block Authentication

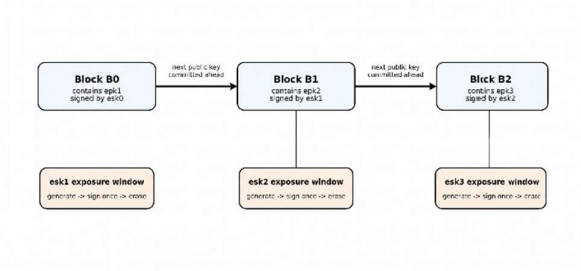

Figure 2. Forward-secure ephemeral-key chaining across consecutive blocks

For consensus operations, each validator maintains a set of ephemeral key pairs. During slot t, validator v_i generates a fresh pair (epk_{i,t+1}, esk_{i,t+1}) for the subsequent slot. The public key epk_{i,t+1} is permanently recorded inside block B_t. When the next slot arrives, the validator utilizes esk_{i,t+1} exactly once to sign the header of B_{t+1} before immediately and securely deleting that specific private key [10].

If a validator is compromised after its private key has been deleted, the attacker can retrieve current or future key material but cannot reconstruct the erased keys from earlier blocks. This approach effectively narrows the vulnerability window and significantly reduces the feasibility of long-range block forgery [3], [11]. While this mechanism provides a highly practical form of forward security suitable for permissioned ledgers, it does not replace formal post-quantum signatures [15] or application-level voting privacy.

4.4. Fault-Tolerant Key Synchronization

This key chain mechanism also introduces a liveness challenge. If a validator misses B_t, it may lack the public key necessary to verify B_{t+1}. To mitigate this risk, validators maintain a short local cache of recently announced public keys. When a required key is missing, the validator requests it from peers and verifies its validity against the relevant block hash before processing it. This bounded retry mechanism is designed to handle moderate network message failures and brief partitions.

4.5. Auditability Through Explicit Signatures

Each committed block carries the explicit cryptographic signatures of a quorum of validators, satisfying the n \ge 3f + 1 safety threshold [6]. Observers with read-only access privileges can autonomously initiate the verification process directly from the genesis block and confirm the absolute correctness of the ledger by verifying the hash chains, block heights, validator signatures, and the chronological continuity of the ephemeral keys. This design approach deliberately avoids relying on an unobservable or internal consensus state as the sole proof of validity, making the distributed ledger significantly more amenable to independent public review and post-election auditing [13].

5. IMPLEMENTATION

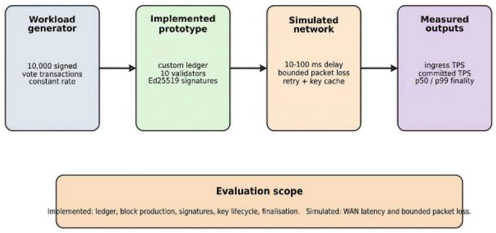

Figure 3. Experimental evaluation workflow and scope

5.1. Software Stack and Deployed Components

The proposed prototype is implemented in the Node.js v20 environment. Its event-driven execution model efficiently supports concurrent transaction dispatch, block propagation, and signature verification without the need to introduce specialized middleware. Digital signatures are implemented using the Ed25519 algorithm via the integrated Node.js crypto module. In terms of memory management, each validator keeps only one active ephemeral private key in working memory and deletes it immediately after successfully signing the next block's header.

Communication between validators relies on HTTP-based peer-to-peer messaging with JSON-encoded packets. Each validator exposes specific endpoints dedicated to submitting transactions, propagating blocks, exchanging signatures, and retrieving public ephemeral keys for the next block. The state of the ledger during the execution of the benchmark test is maintained exclusively in the operational memory and duplicated in the log only for attachment, thus ensuring the reproducibility of the experiment and the possibility of subsequent audit inspection.

Table 1. Implementation scope and evaluation purpose of prototype components

Component	Implemented in prototype		Purpose
Permissioned validator nodes	Yes: 10 independent validator instances		Block proposal, verification, signature exchange, and ledger commit
Append-only ledger	Yes: in-memory state with append-only persistence log		Stores committed blocks and transaction batches
Ephemeral-key lifecycle	Yes: generate, publish, sign once, erase		Enforces bound key exposure and forward security
Network delay/loss	Simulated within		Evaluates behaviour under non-

Component	Implemented in prototype		Purpose
	application logic		ideal wide-area conditions
Client-side ballot secrecy	Out of scope		Assumed to be handled by the voting application layer

5.2. Network Simulation and Benchmark Orchestration

Network effects are simulated within the application logic itself. Each message exchanged between validators was subject to an artificial delay, which was randomly taken from a range of 10 ms to 100 ms, as well as probabilistically bounding packet loss. This segment of the experiment therefore represents a controlled simulation rather than a geographically distributed field application. On the other hand, the blockchain-prototype implementation itself included block creation, transaction grouping, validator signatures, ephemeral key publishing, key erasure, signature verification, and ledger commit — and was fully functional and performed in its native form during the entire benchmark test.

5.3. Hardware Configuration

The experimental evaluation was performed on a standard server configuration equipped with 8 vCPU, 16 GB RAM and Ubuntu Linux 22.04 LTS operating system. We selected this configuration because public-sector systems typically operate with limited infrastructure. While these results should not be interpreted as a guarantee for production deployment, they demonstrate that the proposed mechanism is not inherently too demanding for standard server hardware.

6. Experimental Evaluation

6.1. Experimental Setup and Measurement Scope

The evaluation utilized ten validator instances and a workload of 10,000 signed voting transactions. Our primary objective was to determine whether the ephemeral-key mechanism and explicit signature verification introduce prohibitive computational overhead. Additionally, the experiment provides an initial estimate of throughput and finality, allowing for a direct comparison against the average load requirements of a national-scale election.

6.2. Scope and Evaluation

The evaluation should be understood as a controlled prototype test. It measures the implemented slot workflow, block creation, signature exchange, ephemeral-key publication, key erasure, signature verification, and ledger commit. It also measures behavior under configured latency and message failure.

At the same time, it does not fully reproduce all the real-world properties of an actual election network. The validators were not deployed across geographically diverse data centers or operated by independent institutions. Furthermore, the experiment does not account for authentic inter-institutional routing paths, heterogeneous hardware environments, large-scale DDoS attacks, operational monitoring tools, formal legal certification, or live, voter-facing public infrastructure.

For that reason, the results serve as evidence of feasibility rather than full production validation. A real deployment would require multi-site testing, independent operators, more rigorous adversarial testing, and formal certification.

Table 2. Configuration parameters and metrics for experimental evaluation

Parameter	Value/description
Number of validators	10
Fault threshold assumption	n >= 3f + 1; with n = 10, the design corresponds to f = 3 Byzantine validators
Workload	10,000 signed voting transactions
Network delay	Simulated uniform delay between 10 ms and 100 ms
Primary metrics	Ingress TPS, committed TPS, p50 finality, p99 finality
Implemented blockchain functions	Block production, signature collection, signature verification, ephemeral-key lifecycle, ledger commit
Simulation boundary	Inter-validator latency and bounded message loss

6.3. . Throughput and Finality

The prototype reached a peak in ingress throughput of approximately 583 transactions per second, while the sustained committed throughput was about 274 transactions per second. This difference is entirely expected, as the committed throughput factors in the additional overhead of block creation, consensus message propagation, validator signature collection, verification, and the sequential sign-and-erase procedure.

Finality values were also within a range that appears highly practical for voting. The median finality was 10.7 seconds, and p99 finality stayed below 21 seconds within the simulated network setting. While this is not comparable to high-frequency financial systems that target sub-second settlement times, the more critical requirement in an election context is ensuring that each vote is durably recorded and fully auditable at a later date.

Table 3. Summary of experimental results, observed metrics, and interpretation

Metric	Observed value	Interpretation
Peak ingress throughput	Approx. 583 TPS	The rate at which validator APIs accepted valid transactions
Committed throughput	Approx. 274 TPS	The rate at which transactions were finalised in the ledger

Metric	Observed value	Interpretation
p50 finality	10.7 seconds	Median time from submission to irreversible block inclusion
p99 finality	< 21 seconds	Tail latency under simulated network stress
Cryptographic overhead	Acceptable in the evaluated setting	Ed25519 signing and verification did not become a prohibitive bottleneck

6.4. . Election-Scale Projection

To establish a practical reference point, the measured committed throughput was compared against a straightforward national election scenario involving six million voters and a 13-hour voting window. Under a uniform-load approximation, this setup requires a processing capacity of approximately 128 transactions per second. Consequently, the measured sustained capacity of 274 committed transactions per second yields an average-load safety margin of roughly 2.1.

While that projection is useful, it remains a simplified model. Real-world elections inevitably experience pronounced traffic spikes, particularly during the morning rush and right before polling stations close. Future testing should therefore incorporate non-uniform transaction arrival patterns and be executed across true, geographically distributed infrastructure.

Table 4. National-scale election scenario parameters and prototype validation results

Scenario element	Value
Electorate size	6,000,000 voters
Voting window	13 hours = 46,800 seconds
Average required throughput	Approx. 128 TPS
Measured committed throughput	Approx. 274 TPS
Average-load safety factor	Approx. 2.1
Validation status	Feasibility evidence under controlled prototype and simulated network conditions

7. DISCUSSION

The main takeaway is that the proposed block-level forward security mechanism did not render the prototype too slow for the analyzed average workload. The trade-off is higher latency compared to some generic BFT (Byzantine Fault Tolerant) systems. Within a voting context, however, this compromise can be acceptable, as a finality time of several seconds is usually far less problematic than weak or unreliable audit evidence.

The architecture is intentionally narrower in scope than general-purpose blockchain platforms. It is tailored specifically for use cases where the validators are known, the voting period is strictly bounded, and the long-term integrity of the historical record remains paramount long after the event has concluded. This specific focus explains why each block explicitly carries validator signatures alongside cryptographic information for the subsequent ephemeral public keys.

Ultimately, this evaluation does not prove that the system is ready for a national election; it merely demonstrates that the underlying mechanism is feasible within the tested prototype. Substantiating stronger claims would require side-by-side comparisons with alternative platforms under identical workloads, true distributed deployments, peak-load stress testing, dedicated DDoS experiments, and an independent, third-party security review.

8. LIMITATIONS AND FUTURE WORK

Several open limitations were identified within this research. First, the developed prototype was evaluated on a single regular server running multiple instances of the validator, which implies that a geographically distributed application must be tested separately. Second, network latency and message losses are simulated rather than generated within the actual broadband election network. Third, the projected calculation for six million voters uses the average load model, and in the following stages, it should be expanded with empirical traffic distributions during peak hours. Fourth, the proposed architecture does not solve the issues of vote secrecy, resistance to coercion, or the operation of malicious software (malware) on the client side [13]. Fifth, the system design currently uses the Ed25519 scheme [10] and needs to be evaluated for the integration of post-quantum signature schemes as the technology matures [15].

Future work will therefore primarily focus on distributed multi-site testing, formal ephemeral key machine state verification, comparative analysis with Tendermint [5]/HotStuff-style [17] foundations under equivalent operational loads, integration with ballot privacy protocols [1], [4], [7], [9], as well as detailed evaluation of system behaviour under DDoS attack and node crash recovery scenarios [8].

9. CONCLUSION

This paper presents a feasibility study of a forward-secure (with advanced key security) permissioned blockchain architecture intended for electronic voting at the national level [13], [14]. The proposed architecture successfully combines slot-based block production, explicit multi-validator signatures [6], and a specific ephemeral key mechanism in which public keys are committed one block in advance, while private keys are used once and then permanently deleted [10]. The described design directly addresses a specific risk at the consensus level, which refers to the retroactive falsification of the ledger after the possible compromise of the keys of the validators themselves [3], [11].

The developed custom multi-node prototype demonstrated sustainable commit throughput of approximately 274 TPS and p99 finalization under 21 seconds under simulated broadband network

latency. A projection on an election scenario involving six million voters and a 13-hour voting window provides an average safety factor of approximately 2.1. The stated results provide empirical support for the feasibility of the described approach under the evaluated assumptions, with a simultaneous indication of a clear need for geographically distributed field testing, stronger adversarial evaluation [8], formal security assessment, as well as integration with application-level privacy and verifiability mechanisms [1], [4], [7], [9] before any production deployment in a real environment.

REFERENCES

[1] B. Adida, "Helios: Web-based open-audit voting," in Proc. 17th USENIX Security Symp., 2008, pp. 335–348.

[2] E. Androulaki, A. Barger, V. Bortnikov, et al., "Hyperledger Fabric: A distributed operating system for permissioned blockchains," in Proc. 13th EuroSys Conf., 2018. doi: 10.1145/3190508.3190538.

[3] M. Bellare and S. Miner, "A forward-secure digital signature scheme," in Advances in Cryptology—CRYPTO '99, 1999, pp. 431–448. doi: 10.1007/3-540-48405-1_28.

[4] J. Benaloh, "Verifiable secret-ballot elections," PhD dissertation, Yale Univ., 1987.

[5] E. Buchman, "Tendermint: Byzantine fault tolerance in the age of blockchains," M.S. thesis, Univ. of Guelph, 2016.

[6] M. Castro and B. Liskov, "Practical Byzantine fault tolerance," in Proc. 3rd Symp. Operating Syst. Des. Implementation, 1999, pp. 173–186.

[7] D. Chaum, P. Y. A. Ryan, and S. Schneider, "A practical voter-verifiable election scheme," in European Symp. Res. Comput. Security, 2005, pp. 118–139. doi: 10.1007/11555827_7.

[8] G. Danezis, E. Kokoris-Kogias, A. Sonnino, and A. Spiegelman, "Narwhal and Bullshark: A DAG-based mempool and efficient BFT consensus," in Proc. 31st USENIX Security Symp., 2022, pp. 1971–1988.

[9] L. Harrison, S. Bag, H. Luo, and F. Hao, "VERICONDOR: End-to-end verifiable Condorcet voting with support for strict preference and indifference," ACM Digital Threats: Research and Practice, vol. 5, no. 4, pp. 1–30, 2024. doi: 10.1145/3676267.

[10] G. Itkis and L. Reyzin, "Forward-secure signatures with optimal signing and verifying," in Advances in Cryptology—CRYPTO 2001, 2001, pp. 332–354. doi: 10.1007/3-540-44647-8_20.

[11] H. Krawczyk, "Simple forward-secure signatures from any signature scheme," in Proc. 7th ACM Conf. Comput. Commun. Security, 2000, pp. 108–115. doi: 10.1145/352600.352626.

[12] L. Lamport, R. Shostak, and M. Pease, "The Byzantine generals problem," ACM Trans. Program. Lang. Syst., vol. 4, no. 3, pp. 382–401, 1982. doi: 10.1145/357172.357176.

[13] S. Park, Y. Kwon, and G. Fuchsbauer, "Is blockchain suitable for electronic voting?" IEEE Security & Privacy, vol. 18, no. 1, pp. 70–77, 2020. doi: 10.1109/MSEC.2019.2955709.

[14] M. Pilkington, "Blockchain technology: Principles and applications," in Research Handbook on Digital Transformations, X. Olleros and M. Zhegu, Eds. Edward Elgar, 2016, pp. 225–253.

[15] A. G. Schemitt, H. F. da Silva, R. C. Lunardi, D. Kreutz, R. B. Mansilha, and A. F. Zorzo, "Assessing the impact of post-quantum digital signature algorithms on blockchains," arXiv:2510.09271, 2025.

[16] M. Swan, Blockchain: Blueprint for a New Economy. Sebastopol, CA: O'Reilly Media, 2015.

[17] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, "HotStuff: BFT consensus with linearity and responsiveness," in Proc. 2019 ACM Symp. Principles Distrib. Comput., 2019, pp. 347–356. doi: 10.1145/3293611.3331591.

[18] D. Vukmirović, S. Radojičić, N. Stanojević, N. Dragović, and T. Čomić, "Enhancing critical infrastructure resilience: An integrated framework for hazardous material management of electric vehicle batteries," in Critical Infrastructure: Risk and Crisis Management in the New Reality, Z. Keković, S. Kentera, M. Dabetić, and S. Grbović, Eds. Springer, 2026, pp. 81–109. doi: 10.1007/978-3-032-20948-1_5.ž.

Nebojša Stanojević holds a master's degree in Geography, is a PhD student at the Faculty of Organizational Sciences, University of Belgrade, and works as a high school teacher and a data analyst in automotive parts distribution. He is also the founder and software architect at INFOS-N, an IT services agency, where he has extensive experience in designing and implementing numerous domestic and national-scale projects. His research and professional interests are mainly focused on blockchain technology, geographic information systems (GIS), data analytics, and the application of generative artificial intelligence in business and education. E-mail: ns20205003@student.fon.bg.ac.rs, ORCID: 0009-0004-4418-0039.

Dragan Vukmirović is a full professor at the Faculty of Organizational Sciences, University of Belgrade, in the fields of statistics, data science, e-business, and digital transformation. He is a former Director General of the Statistical Office of the Republic of Serbia and has extensive experience in official statistics, statistical methodology, information systems, and evidence-based decision-making. He is the author and co-author of numerous scientific and professional papers and has participated in national and international projects. His research interests include statistics, data science, e-government, digital inclusion, artificial intelligence, and the application of generative AI in education, business, and public administration. E-mail: dragan.vukmirovic@fon.bg.ac.rs, ORCID: 0000-0003-0248-016X.

Tijana Čomić is an assistant professor and statistician with a PhD in Statistics and extensive experience in household surveys, official statistics, and international research projects. Her professional work includes survey methodology, data analysis, Multiple Indicator Cluster Surveys (MICS), EU-SILC, GEDSI indicators, and evidence-based policy support. She has worked as an international consultant and has contributed to research and capacity-building activities in the field of statistics and development. Her recent work also includes the practical application of generative artificial intelligence and educational workshops in this emerging field. Her research interests include statistics, survey methodology, social indicators, Big Data, and generative AI. E-mail: tijana.comic@vos.edu.rs, ORCID: 0000-0001-6556-5879.

Dejan Stanojević is an Information Systems Engineer who graduated from the Faculty of Organizational Sciences, University of Belgrade. He has more than 35 years of professional experience in the development of complex business, geospatial, analytical, and decision-support systems for public administration, utilities, banking, telecommunications, healthcare, pharmaceutical analytics, and automotive parts distribution. He is the software architect at InSist doo and has worked remotely on international projects for European and North American companies and clients, with a particular focus on the architecture and implementation of large-scale information systems.

Miloš Cetković holds a degree in Management and a master's degree in Business Analytics from the Faculty of Organizational Sciences, University of Belgrade. His academic and research interests focus on retail development

strategies, omnichannel business models, digital transformation, and the competitiveness of physical stores in the contemporary e-commerce environment. He has more than 25 years of professional experience in trade, retail, and international business. His career includes founding a company for the import and wholesale of computer equipment and developing a retail network of ten physical stores and four online sales channels in Serbia, Montenegro, and Bosnia and Herzegovina. E-mail: Milos@ekspedicija.rs, ORCID: 0009-0000-9537-4651.