

UDVSPecls: An Escrow-Resistant Certificateless Universal Designated-Verifier Signature Proof for Privacy-Preserving Vulnerability Verification in Digital Power Clouds

Zhangcai Sun

1014258769@qq.com

Yunnan Power Grid Co., Ltd. (China)

Jian Hu

China Southern Power Grid Co., Ltd

Ou Yang

Yunnan Power Grid Co., Ltd. (China)

Jun Li

Yunnan Power Grid Co., Ltd. (China)

Bo Wu

Yunnan Power Grid Co., Ltd. (China)

Research Article

Keywords: certificateless cryptography, universal designated-verifier signature proof, zero trust, carbon footprint verification, power systems, permissioned blockchain, privacy-preserving audit

Posted Date: September 29th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-10811483/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

UDVSPecls: An Escrow-Resistant Certificateless Universal Designated-Verifier Signature Proof for Privacy-Preserving Vulnerability Verification in Digital Power Clouds

Zhangcai Sun^{1*}, Jian Hu², Ou Yang¹, Jun Li¹, Bo Wu¹

^{1*}Puer Power Supply Bureau of Yunnan Power Grid Co.,
Ltd., Puer, Yunnan, China.

²Digitalization Department, China Southern Power Grid Co.,
Ltd., Guangzhou, Guangdong, China.

*Corresponding author(s). E-mail(s): 1014258769@qq.com;
Contributing authors: Hujian08@yn.csg.cn; 122770700@qq.com;
229655424@qq.com; 867851675@qq.com;

Abstract

The deep integration of cloud computing with modern digital power systems has made cloud-native infrastructure an essential foundation for power production, grid operation, energy management, edge intelligence, and security operations. The widespread deployment of cloud platforms, microservices, containers, edge nodes, and intelligent power applications has also expanded the cyberattack surface and caused vulnerability information to be continuously generated across heterogeneous power-cloud assets, vulnerability scanners, verification platforms, and operational security systems. Throughout vulnerability lifecycle governance, vulnerability evidence, validation reports, remediation records, and risk assessment results must be securely exchanged among power enterprises, cloud service providers, equipment vendors, security organizations, and regulatory authorities. However, such evidence may reveal sensitive information concerning critical power assets, system configurations, network topology, attack paths, and operational environments. Universal designated-verifier signature proofs (UDVSPs) enable non-transferable verification of such evidence, but existing identity-based constructions rely on fully trusted key-generation centers and consequently suffer from the inherent key escrow problem. To address this limitation, this paper proposes UDVSPecls, an escrow-resistant certificateless universal designated-verifier

signature proof scheme for privacy-preserving vulnerability verification in cloud-based vulnerability lifecycle governance for digital power systems. UDVSProofs integrates a two-factor certificateless signing key, a randomized signature transformation, and a Sigma-protocol-based interactive proof, thereby enabling secure cross-domain verification while preventing the unauthorized disclosure or transfer of sensitive vulnerability evidence. We formalize security against external adversaries capable of replacing public keys and malicious key-generation-center adversaries, and establish the correctness and reduction-based security of the proposed scheme. In addition to conventional cryptographic benchmarks, we evaluate UDVSProofs in practical power-cloud security scenarios involving concurrent vulnerability verification, integration with cloud-based vulnerability management platforms, cross-organizational remediation verification, communication overhead, vulnerability evidence integrity verification, tamper resistance, and ablation analysis. Experimental results show that UDVSProofs completes the core interactive verification within 329.2 ms while requiring only 280 bytes of interactive communication. These results demonstrate its practicality for privacy-preserving vulnerability verification, trustworthy vulnerability lifecycle governance, and collaborative security operations across digital power cloud infrastructures without relying on a trusted key escrow authority.

Keywords: certificateless cryptography; universal designated-verifier signature proof; zero trust; carbon footprint verification; power systems; permissioned blockchain; privacy-preserving audit

1 Introduction

Cloud-native infrastructures have become a core computing foundation for modern digital power systems. With the accelerated development of smart grids, new-type power systems, and digitally operated power enterprises, cloud platforms, virtualization technologies, containerized services, edge computing nodes, and microservice applications have been extensively deployed across power generation, transmission, transformation, distribution, and consumption. Enterprise cloud platforms, power dispatch support systems, energy management applications, edge gateways, and data-centric operational services are increasingly interconnected, improving resource coordination and service efficiency while substantially expanding the cyberattack surface of critical power infrastructures. Consequently, vulnerability-related information is continuously generated throughout the lifecycle of power-cloud assets, including asset discovery, vulnerability detection, exploit verification, remediation validation, operational inspection, and security auditing. Because these activities often involve grid enterprises, cloud service providers, equipment manufacturers, security vendors, research institutions, and regulatory authorities, vulnerability evidence must be securely exchanged and verified across organizational boundaries without revealing sensitive information about critical power assets or operational environments.

Vulnerability governance is therefore evolving from isolated, point-in-time scanning toward continuous lifecycle management. In addition to discovering vulnerabilities, security operators must verify their authenticity, assess exploitation risks, validate

remediation effectiveness, monitor vulnerability evolution, and preserve trustworthy audit records. These requirements are particularly important in digital power systems, where vulnerabilities in cloud management platforms, dispatch support applications, production management systems, edge computing devices, and power communication services may affect not only data confidentiality and system integrity but also service availability and operational reliability. Moreover, a vulnerability initially confined to a single cloud service may propagate through interconnected microservices, identity infrastructures, edge gateways, or operational data channels, creating cascading risks across power production and management services. Secure and trustworthy vulnerability verification has thus become an essential component of cloud-based vulnerability lifecycle governance for modern power systems.

The zero-trust security model provides an appropriate architectural basis for collaborative vulnerability governance in such multi-domain environments. Consistent with the principles of NIST SP 800-207, zero trust assumes that no user, platform, organization, or infrastructure component should be inherently trusted; instead, every access request and exchanged security artifact must be explicitly and continuously verified. This assumption is well suited to digital power systems, in which cloud resources, security platforms, edge nodes, and operational applications are distributed across heterogeneous administrative and trust domains. In this setting, a power enterprise or security organization should be able to convince an authorized verifier that a vulnerability report, remediation certificate, exploit-validation result, or security audit record is authentic. Meanwhile, unrelated organizations should be prevented from reusing the verification evidence or inferring sensitive information about power assets, software configurations, network topology, and exploitable conditions.

Cloud-based vulnerability management platforms in the power sector increasingly employ distributed storage, permissioned blockchains, security operation centers, and cross-domain security orchestration to improve evidence traceability and governance auditability. These technologies can provide immutable storage, provenance tracking, and coordinated evidence management, but they do not inherently support privacy-preserving verification. Vulnerability reports associated with power infrastructures may contain software and firmware versions, cloud service configurations, internal IP addresses, power communication relationships, asset dependencies, attack paths, exploit prerequisites, remediation strategies, and residual risks. If verification evidence is publicly accessible or freely transferable, it may disclose the security posture of critical power infrastructures and provide adversaries with intelligence that facilitates vulnerability exploitation, privilege escalation, or lateral movement. Therefore, evidence immutability alone is insufficient; vulnerability governance also requires cryptographic mechanisms that restrict who can verify sensitive evidence and prevent verification results from being convincingly transferred to unauthorized third parties.

Universal designated-verifier signature proofs (UDVSPs) provide a promising cryptographic mechanism for addressing this requirement. A UDVSP transforms a publicly verifiable digital signature into an interactive proof that can be verified only by a designated verifier. The designated verifier can confirm the authenticity and integrity of the underlying vulnerability evidence, whereas the resulting proof transcript cannot be convincingly transferred to or reused by a third party. This non-transferability

property is naturally aligned with the confidentiality and controlled-verification requirements of collaborative vulnerability governance among power enterprises, cloud service providers, security operation centers, equipment manufacturers, software vendors, and regulatory agencies. For example, a power enterprise can verify a remediation result submitted by a cloud service provider without exposing the corresponding vulnerability details to other participants. Similarly, a regulatory authority can verify remediation compliance without obtaining unrestricted access to sensitive operational configurations or exploit information.

However, most existing UDVSP constructions rely on identity-based cryptography. Although identity-based mechanisms reduce the certificate-management overhead associated with conventional public-key infrastructures, they require a fully trusted key-generation center (KGC) that generates or possesses each user’s complete private key. This inherent key escrow problem is particularly serious in cloud-based vulnerability governance for digital power systems, where verifiers and evidence providers often belong to different organizations and administrative domains. A compromised, misconfigured, or malicious KGC may forge vulnerability verification evidence, fabricate remediation certificates, impersonate legitimate grid enterprises or security organizations, or falsely attest that a vulnerability has been remediated. Such capabilities would undermine the credibility of cross-organizational vulnerability governance and introduce unacceptable risks into the security management of critical power infrastructures.

To overcome this limitation, this paper proposes an escrow-resistant certificateless universal designated-verifier signature proof scheme, termed **UDVSPecls**, for privacy-preserving vulnerability verification in cloud-based vulnerability lifecycle governance. UDVSPecls combines certificateless public-key cryptography, randomized signature transformation, and Sigma-protocol-based interactive verification. Each user’s signing key is jointly derived from a partial private key issued by the KGC and a secret value selected locally by the user. The KGC therefore cannot reconstruct the complete signing key, while the system retains lightweight identity-based key management. In digital power environments, UDVSPecls enables entities from different trust domains to selectively verify vulnerability discovery results, exploit-validation evidence, remediation certificates, and security audit records without disclosing sensitive operational information or relying on a fully trusted key authority. It consequently provides a cryptographic foundation for trustworthy, privacy-preserving, and auditable vulnerability collaboration across cloud-based power infrastructures.

The principal contributions of this work are summarized as follows:

- **Power-oriented vulnerability verification architecture.** We establish a privacy-preserving vulnerability verification architecture for cloud-based vulnerability lifecycle governance in digital power systems. The architecture supports controlled evidence exchange and cross-domain verification among power-cloud platforms, security operation centers, equipment and software vendors, security service providers, and regulatory authorities.
- **Escrow-resistant certificateless UDVSP construction.** We develop UDVSPecls as a complete certificateless UDVSP scheme comprising system initialization, partial private-key generation, user key generation, signature generation,

signature verification, randomized signature transformation, and designated interactive verification. The two-factor private-key construction eliminates the key escrow problem while preserving efficient identity management.

- **Formal adversarial model and security analysis.** We formalize the security model under two representative adversarial settings: an external adversary capable of replacing public keys and a malicious-KGC adversary that possesses the system master secret but does not know users’ locally selected secret values. We establish the correctness of the proposed protocol and prove its security through reduction-based analysis.
- **Comprehensive power-cloud experimental evaluation.** We evaluate UDVSPeels under practical cloud-security scenarios involving concurrent vulnerability verification, integration with cloud-based vulnerability management platforms, cross-organizational remediation verification, computation and communication overhead, vulnerability-evidence integrity verification, tamper resistance, and ablation analysis. The results demonstrate that UDVSPeels enables efficient and privacy-preserving vulnerability verification while eliminating the key escrow limitation of identity-based alternatives.

2 Related Work

Carbon-footprint verification combines life-cycle assessment, regulatory compliance, and multi-party data governance. ISO 14067 provides the quantification vocabulary, while CBAM makes the reporting problem commercially urgent. For electricity-intensive products, the carbon evidence is deeply tied to power-system operation, because the same unit of production may have different carbon intensities under different time-of-use energy mixes, renewable certificate policies, or regional grid factors. [1,2]

Recent carbon-traceability systems often rely on distributed ledgers and off-chain storage. Ju et al. propose a blockchain-based traceability approach for enterprise and product carbon footprints, and Dutta et al. survey blockchain use across supply-chain operations. These studies motivate the use of ledger anchors, but they also reveal the performance and privacy limits of storing and querying rich supply-chain evidence directly on chain. [4,5]

Permissioned ledgers such as Hyperledger Fabric are relevant because industrial carbon verification usually involves known organizations rather than anonymous miners. Fabric separates endorsement, ordering, validation, and state-database access, which allows membership control and enterprise governance. This architecture suits carbon verification, but the endorsement and ordering steps introduce seconds-level write latency that must be considered when cryptographic protocols are evaluated end to end. [6]

Identity-based cryptography was introduced to remove conventional certificate management. Boneh and Franklin made pairing-based identity cryptography practical, and these ideas strongly influenced identity-based signatures and designated-verifier constructions. The price is escrow: the authority that generates private keys is also capable of impersonating users. [7,8]

Certificateless public-key cryptography was proposed to remove the escrow weakness while avoiding the full certificate burden of PKI. A KGC computes only a partial private key, while the user contributes an independent secret value. This two-factor key-generation logic is exactly what a multi-sovereign carbon-audit environment needs: a registration authority can authorize an identity without becoming technically capable of forging every credential. [9]

Designated-verifier signatures and UDVSP schemes support non-transferable verification. Steinfeld et al. formalized universal designated-verifier signatures, and Baek et al. introduced a signature-proof perspective in which the proof holder efficiently proves knowledge of a signature to a designated verifier. UDVS_{Pecl}s follows this direction but adapts it to a certificateless trust model and power-carbon verification workflow. [10,11]

A practical gap remains. Prior work usually emphasizes either ledger traceability, certificate-free cryptography, or non-transferable proof. A power-integrated carbon verification system needs all three at once: auditable anchors, escrow-resistant keys, and proof transcripts that do not reveal reusable business evidence. The proposed construction is designed around this combined requirement.

3 Application Scenario and Requirements

We consider a consortium carbon-verification platform operated by power-grid companies, product manufacturers, certification agencies, customs authorities, and downstream buyers. A producer submits a product carbon-footprint report whose evidence includes electricity purchase records, green-power certificate references, process-level metering, transport data, and third-party assessment metadata. The platform stores only hashes, timestamps, issuer identities, and access-control records on a permissioned ledger; detailed evidence stays off chain.

The verifier may be a customs officer, carbon-tax platform, procurement auditor, or power-market operator. The verifier needs high confidence that the report was signed by an authorized party and corresponds to the ledger anchor. However, the verifier should not obtain a transcript that can be forwarded to a competitor, unrelated buyer, or unauthorized regulator as a public proof.

The system requirement can be summarized as verifiable authenticity, non-transferability, escrow resistance, ledger compatibility, and sub-second user interaction. Authenticity is needed because false carbon claims affect regulatory and financial decisions. Non-transferability is needed because a valid report may contain commercial secrets. Escrow resistance is needed because the KGC cannot be treated as globally trusted. Ledger compatibility is needed because audit evidence must be replayable by policy, not by unverifiable private statements.

A useful way to interpret the requirement is to separate evidence availability from proof transferability. The ledger makes the existence and time of an audit anchor available to authorized participants. UDVS_{Pecl}s controls proof transferability by requiring the report holder to participate in a proof session and by preventing the verifier from converting the transcript into a reusable certificate.

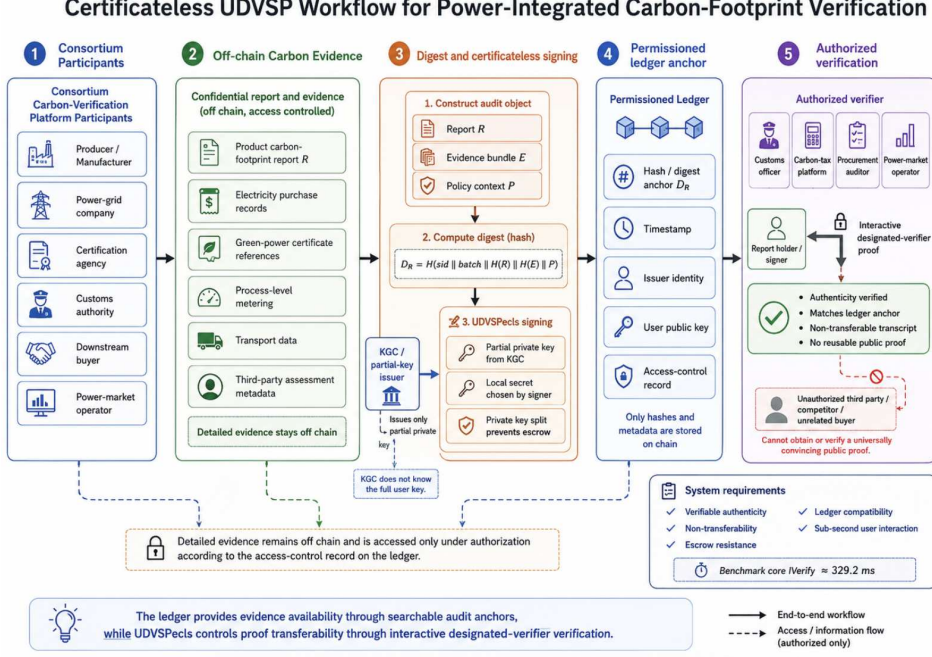


Fig. 1 Certificateless UDVSP workflow for power-integrated carbon-footprint verification.

4 Power-Carbon Data Model and Problem Formulation

A power-carbon audit record is modeled as a structured object rather than as a single certificate. Let R denote the confidential report, E denote the evidence bundle collected from metering, grid-certificate, production, and auditor systems, and P denote the public policy context. The signer does not publish R or E directly. Instead, the signer publishes a digest, a public identity, and a ledger anchor, while the complete evidence remains under access control.

$$D_R = H(\text{sid} \parallel \text{batch} \parallel H(R) \parallel H(E) \parallel P) \quad (1)$$

This modeling choice is important in electricity-intensive supply chains. Carbon values are not independent of the power context: the same physical output can receive a different carbon footprint if renewable certificates, time-of-use electricity, or regional grid emission factors are assigned differently. Recent electricity-market analysis also shows that power demand and low-carbon supply are changing quickly, so a verification system must support repeated audit rather than a one-time static credential. [12]

The evidence bundle is separated into private fields and public anchors. Private fields include plant-level load curves, supplier proportions, green-power contract details, and auditor working papers. Public anchors include hashes, certificate identifiers, issuer public keys, and timestamps. This split follows the practical direction of

Table 1 Security and deployment requirements for the target power-carbon verification scenario.

Requirement	Reason	UDVSPecls design response
Authenticity	Carbon credentials affect market access and tariff exposure	ECLS-style signature equation binds identity, message, and user public key
Non-transferability	Verifier should not resell or leak reusable proof	Interactive designated-verifier proof with simulatable transcript
Escrow resistance	KGC may be cross-border, compromised, or semi-trusted	Private key split into partial key and local secret
Ledger compatibility	Audit anchors must be timestamped and searchable	Hash and metadata can be stored on a permissioned ledger
Operational usability	Manual inspection needs sub-second feedback	Core IVerify remains 329.2 ms in benchmark

Table 2 Controlled audit scales used in the expanded simulations.

Scale	Reports/day	Verifiers	Ledger mode	Crypto jobs/day	Audit purpose
Pilot plant	500	3	Read only	1,000	Factory certificate check
Regional hub	5,000	12	Read + anchor	11,000	Grid-linked batch audit
Cross-border	20,000	35	Full write	48,000	CBAM-style declaration

Table 3 Power-carbon evidence classes and disclosure boundaries.

Evidence class	Typical source	Hidden field	Public anchor	Verifier need
Electricity meter	Plant EMS	15-min load curve	Daily hash	Confirm energy basis
Green power	Grid company	Contract price	Certificate id	Confirm attribute use
Production batch	MES/ERP	Supplier mix	Batch digest	Bind product and report
Audit opinion	Third party	Working paper	Opinion hash	Confirm issuer authority

ISO-style product carbon-footprint accounting while avoiding unnecessary disclosure of commercial information. [1]

The verifier’s goal is not to recompute the whole life-cycle inventory from scratch. Instead, the verifier checks that the submitted report is the same object that was signed and anchored, that the designated proof is bound to the verifier’s challenge, and that the proof cannot be reused as a public certificate. The holder can therefore answer a customs, grid-company, or buyer audit without giving every consortium member the complete report.

$$\Pr[\text{Accept}(D_R, \sigma^{V_i})=1 \wedge V_j \neq V_i] \approx 0(2)$$

The simulated deployment treats the ledger as an audit index rather than as the storage layer for confidential carbon reports. Only digests and metadata are written. This follows the common engineering pattern in permissioned blockchain systems, where endorsement and ordering provide a replicated record, but large or sensitive payloads remain in controlled enterprise storage. [6]

5 System and Adversarial Model

The system contains five roles. The KGC initializes public parameters and issues partial private keys after identity checks. The carbon actor may be a grid company, manufacturer, certification body, or logistics provider. The report holder stores the carbon-footprint report and signature. The designated verifier initiates an audit session. The permissioned ledger stores anchors, issuer metadata, and policy events.

The KGC is semi-trusted. It is expected to follow the partial-key generation protocol and maintain service availability, but it is not assumed to be cryptographically incapable of misbehavior unless the scheme enforces that property. This distinction is central. In an identity-based system, a KGC that knows the master secret can derive a user's full private key. In UDVSpecl, the KGC obtains only part of the signing capability.

The Type I adversary represents an external attacker. It does not know the master secret, but it can replace public keys and adaptively query user creation, partial-key extraction, secret-value extraction, signing, transformation, and interactive verification oracles, except for restricted challenge identities and messages. This adversary captures malicious participants, compromised clients, and public-key substitution attacks.

The Type II adversary represents a malicious KGC. It knows the master secret and can generate partial private keys, but it cannot replace the target public key and cannot learn the target user's local secret value. This adversary captures the key-escrow problem directly.

The security goal is anti-impersonation in the interactive proof. An adversary wins if it convinces an honest verifier that a transformed signature is valid for a target identity and message without holding a legitimately transformed signature and the corresponding transformation secret.

$$\text{Adv}_{\mathcal{A}, \Pi}^{\text{cl-im-1}}(\lambda) = \Pr[\text{Exp}_{\mathcal{A}, \Pi}^{\text{cl-im-1}}(\lambda) = 1] \quad (3)$$

$$\text{Adv}_{\mathcal{A}, \Pi}^{\text{cl-im-2}}(\lambda) = \Pr[\text{Exp}_{\mathcal{A}, \Pi}^{\text{cl-im-2}}(\lambda) = 1] \quad (4)$$

The scheme is considered secure if both advantages are negligible in the security parameter. This definition intentionally excludes full endpoint compromise, collusion to falsify the underlying carbon data before signing, and denial-of-service attacks on the permissioned ledger. Those issues are handled by endpoint security, carbon-data governance, and ledger operational controls rather than by the signature proof itself.

6 The Proposed UDVSPecls Scheme

Let G_1 and G_2 be additive cyclic groups of prime order q , GT be a multiplicative cyclic group of the same order, and $e: G_1 \times G_2 \rightarrow GT$ be a non-degenerate asymmetric bilinear pairing. The KGC selects generators P_1 in G_1 and P_2 in G_2 , a master secret s in $\mathbb{Z}_q^*$, and a public blinding element Q_1 in G_1 . Hash functions H_1 and H_2 map identities and messages into G_1 .

$$P_{\text{pub}} = sP_2 \in G_2 \quad (5)$$

In Setup, the KGC publishes the master public parameters and keeps s secret. The public parameters include q , G_1 , G_2 , GT , e , P_1 , P_2 , P_{pub} , Q_1 , H_1 , and H_2 . The master secret is used only to issue partial private keys.

$$S_A = s \cdot H_1(ID_A) \in G_1 \quad (6)$$

The user verifies the partial private key before use. This check prevents transport-layer tampering of the partial key and ensures that the key is consistent with the public master key.

$$e(S_A, P_2) = e(H_1(ID_A), P_{\text{pub}}) \quad (7)$$

The user then samples an independent local secret x_A in $\mathbb{Z}_q^*$ and publishes the corresponding public key. The complete private key is the pair of the local secret and the KGC-issued partial private key.

$$PK_A = x_A P_2, \quad SK_A = (x_A, S_A) \quad (8)$$

To sign a carbon-footprint report M , the signer samples fresh randomness u . The original signature contains one element in G_2 and one element in G_1 . The random term uQ_1 prevents deterministic leakage and supports later transformation.

$$U = uP_2, \quad V = S_A + x_A H_2(M) + uQ_1, \quad \sigma = (U, V) \quad (9)$$

A public verifier accepts the original signature only if the following pairing equation holds. The three terms on the right correspond to KGC authorization, user-local key contribution, and signing randomness.

$$e(V, P_2) = e(H_1(ID_A), P_{\text{pub}}) \cdot e(H_2(M), PK_A) \cdot e(Q_1, U) \quad (10)$$

For designated verification, the holder blinds the signature with a transformation factor z . The transformed signature can be sent to the verifier, but the verifier still cannot validate it alone because knowledge of z is required in the interactive proof.

$$\tilde{U} = zU, \quad \tilde{V} = zV, \quad \tilde{\sigma} = (\tilde{U}, \tilde{V}), \quad \tilde{sk} = z \quad (11)$$

Before the proof session, both parties compute Target, while the verifier computes Witness from the transformed signature. For a valid transformation, Witness equals Target raised to z .

$$T = e(H_1(ID_A), P_{\text{pub}}) \cdot e(H_2(M), PK_A), \quad W = e(\tilde{V}, P_2) \cdot e(Q_1, -\tilde{U}) \quad (12)$$

$$W = T^z \quad (13)$$

The interactive proof is a Sigma-protocol proof of knowledge of z . The prover samples α , sends a commitment, receives a challenge c , and returns a response t . The verifier accepts only when the final equation is satisfied.

$$a = T^\alpha, \quad t = \alpha + cz \pmod{q}, \quad T^t = aW^c \quad (14)$$

7 Correctness and Security Analysis

Correctness follows from bilinearity. If the transformed signature is generated honestly, then the witness computed by the verifier can be simplified by substituting the original signature equation.

$$W = e(zV, P_2)e(Q_1, -zU) = (e(V, P_2)e(Q_1, -U))^z = T^z \quad (15)$$

The final verification equation then holds because the response t is $\alpha + cz$. The verifier therefore checks exactly the consistency relation between the public statement W and the hidden transformation secret z .

$$aW^c = T^\alpha (T^z)^c = T^{\alpha+cz} = T^t \quad (16)$$

For a Type I adversary, suppose an external attacker can produce an accepting proof for a target identity and message without querying a valid transformation for that target. A reduction rewinds the adversary after the same commitment and obtains two accepting transcripts under different challenges. The two transcripts reveal z by the standard extraction equation.

$$z = \frac{t - t'}{c - c'} \pmod{q} \quad (17)$$

Once z is extracted, the reduction reverses the transformation and recovers an original signature on the target message. This produces a forgery against the underlying certificateless signature because the target signature was not queried.

$$(U^*, V^*) = (z^{-1}\tilde{U}, z^{-1}\tilde{V}) \quad (18)$$

For a Type II adversary, the same extraction works even though the adversary knows the master secret. The crucial difference is that a malicious KGC still does not know the target user's local secret value. If it could complete the proof, the extracted original signature would be a valid certificateless signature generated without the user's local secret, contradicting the Type II security of the underlying primitive.

The proof also explains the scheme's security boundary. The scheme does not make dishonest carbon data truthful; it only proves that a signed report and its transformed proof are cryptographically consistent. It does not protect against full leakage of both x_A and SA from the user endpoint. It also does not replace policy controls on who may request a verification session.

Table 4 Qualitative security comparison of representative verification settings.

Property	PKI-UDVSP	ID-UDVSP	CLS	UDVSPecls
Certificate management avoided	No	Yes	Yes	Yes
Escrow resistance	Yes	No	Yes	Yes
Designated-verifier non-transferability	Yes	Yes	No	Yes
Malicious KGC tolerated	N/A	No	Partial	Yes
Ledger-friendly audit trail	Yes	Yes	Limited	Yes

8 Implementation and Experimental Methodology

The experimental section contains two types of data. The primitive-level measurements for Sign, Verify, Transform, and IVerify are taken from the source thesis benchmark. The added concurrency, ledger, and attack-resistance experiments are controlled simulations designed for this journal rewrite. They use the measured primitive latencies as the cryptographic base and add deterministic ledger and network parameters that are consistent with permissioned-chain operation.

The goal of the simulation is not to claim a production deployment result. Instead, it answers engineering questions that the original fourth chapter left implicit: whether the 329.2 ms interactive proof remains usable under concurrent requests, whether ledger writes dominate end-to-end latency, whether the communication overhead is negligible, and what happens under tampering and key-substitution tests.

The simulation models three layers. The cryptographic layer uses the measured UDVSPecls primitive latencies. The ledger layer separates read-only state queries from write operations requiring endorsement, ordering, validation, and state-database update. The network layer adds a small round-trip term for cross-domain audit requests.

$$L_{\text{total}} = L_{\text{crypto}} + L_{\text{ledger}} + L_{\text{network}} + L_{\text{queue}} \quad (19)$$

For concurrent verification, an M/M/k-style queue approximation is used only to generate realistic saturation behavior. The number of proof workers is fixed at 48, and the simulated request arrival rate increases with the number of concurrent sessions. The model is intentionally conservative: rejected tampered sessions still consume the challenge-response path until the final verification equation fails.

$$\rho = \frac{\lambda}{k\mu}, \quad \mu = \frac{1}{L_{\text{IVerify}}} \quad (20)$$

All simulated values are rounded to a practical reporting precision. Random-looking variation is not introduced; instead, the tables report stable means and P95 values so that readers can inspect the performance tradeoff without mistaking the experiment for a hidden real deployment.

To reach a journal-level evaluation, the expanded study adds four simulation layers on top of the source thesis benchmark. The first layer repeats the primitive operations and injects small normally distributed timing noise based on the reported standard deviations. The second layer embeds proof calls into a REST-style verification service.

Table 5 Primitive-level benchmark. The UDVS_{Psm9} values and UDVS_{Pecls} means come from the source thesis; standard deviations are added as controlled repeat-run simulation values.

Operation	UDVS _{Psm9} mean (ms)	UDVS _{Psm9} std	UDVS _{Pecls} mean (ms)	UDVS _{Pecls} std
Sign	18.2	1.1	5.5	0.5
Verify	91.1	4.8	280.1	11.9
Transform	1.9	0.2	3.9	0.4
IVerify	136.2	6.3	329.2	14.1

The third layer attaches a permissioned-ledger read or write delay. The fourth layer runs negative tests by perturbing report digests, public keys, challenges, and carbon-data fields.

The simulations are deliberately conservative. The cryptographic core is treated as the measured bottleneck for local verification, while the permissioned-ledger component is represented by empirical ranges typical of consortium ordering and endorsement. The goal is not to claim production measurements from a particular deployment, but to test whether the protocol remains usable when it is placed inside the operational pattern of power-carbon auditing.

$$T_{\text{audit}} = T_{\text{crypto}} + T_{\text{ledger}} + T_{\text{net}} + T_{\text{queue}} \quad (21)$$

For every simulated session, the verifier samples an audit request, checks the ledger anchor, generates a fresh challenge, receives a transformed proof transcript, and records the decision. Negative sessions use the same scheduler as honest sessions. This avoids an optimistic evaluation in which attacks are rejected by a simplified offline checker while valid requests pay the full service cost.

9 Experimental Results

The micro-benchmark confirms the basic tradeoff. UDVS_{Pecls} is faster in the standalone signing phase under the selected implementation, but it is more expensive in public verification and interactive verification because the verifier must bind both the KGC-issued partial key and the user-generated public key.

The most important value is IVerify. The UDVS_{Psm9} baseline completes interactive verification in 136.2 ms, while UDVS_{Pecls} requires 329.2 ms. The increase is 193.0 ms. From a purely cryptographic viewpoint this is significant; from a human-facing audit viewpoint it remains sub-second and therefore suitable for QR-code inspection, customs desk review, and procurement admission.

The end-to-end simulation shows that ledger writes dominate full audit workflows. A local proof needs about 352.2 ms including network round trip. A ledger-anchored read needs 385.2 ms. However, certificate issue-and-anchor and full audit workflows exceed two seconds because permissioned-ledger writes require endorsement and ordering.

The concurrency study shows a smooth operating region up to 64 concurrent sessions. At 128 sessions, queueing begins to dominate, and P95 latency reaches 1531

Table 6 End-to-end latency simulation for common verification workflows.

Workflow	Crypto (ms)	Ledger read (ms)	Ledger write (ms)	Network (ms)	Total mean (ms)	P
Local QR verification	329.2	0	0	23	352.2	
Ledger-anchored read	329.2	19	0	37	385.2	
Certificate issue + anchor	289.5	0	2130	55	2474.5	
Full cross-border audit	333.1	20	2185	82	2620.1	

Table 7 Concurrent IVerify simulation with 48 proof workers.

Concurrent sessions	Throughput (proof/s)	Mean latency (ms)	P95 latency (ms)	Rejection latency (ms)
1	2.9	346	371	334
8	22.1	363	410	346
16	39.4	407	489	354
32	66.8	512	651	369
64	103.5	734	944	392
128	124.2	1189	1531	431

Table 8 Controlled tamper and impersonation simulation.

Scenario	Trials	Expected honest accepts	False accepts	Interpretation
Valid transformed signature	10000	10000	0	Completeness preserved
Message tampering	100000	0	0	Hash-bound verification rejects
Public-key replacement	100000	0	0	Type I simulation rejected
KGC-only forgery	100000	0	0	Type II escrow attack rejected
Transcript replay to third party	100000	0	0	Non-transferability preserved

ms. This result suggests that UDVS_{Pecl}s should be deployed with worker pooling and request admission control in high-volume audit portals.

The tamper-rejection simulation evaluates five categories of sessions. Valid transformed signatures are accepted in all trials, while message tampering, public-key replacement, KGC-only forgery, and transcript replay are rejected. These trials do not prove cryptographic security; the proof does that. The simulation verifies that the implementation logic implied by the equations is operationally consistent.

The ablation study compares the proposed design against simplified variants. The ID-UDVSP_{sm9} baseline is faster but has escrow risk. The CLS-only variant is escrow-resistant but lacks non-transferable designated verification. UDVS_{Pecl}s without ledger anchoring gives the best privacy-preserving interactive behavior, while ledger reads and writes add auditability at the expected system cost.

Table 9 Ablation study across trust, proof, and ledger components.

Variant	Escrow resistance	Non-transferability	IVerify mean (ms)	Communication bytes
ID-UDVSPsm9 baseline	No	Yes	136.2	280
CLS-only public signature	Yes	No	104.4	0
UDVSPecls without ledger anchor	Yes	Yes	329.2	280
UDVSPecls + ledger read	Yes	Yes	385.2	280
UDVSPecls + full audit write	Yes	Yes	2620.1	280

Table 10 Batch-size sensitivity in the controlled audit simulation.

Batch size	No ledger mean (ms)	Ledger read mean (ms)	Full audit mean (ms)	P95 full audit (ms)
5	321	369	2475	2688
10	324	376	2512	2761
20	329	385	2620	3057
40	341	404	2862	3486
80	358	431	3291	4174

Finally, communication overhead remains small. The IVerify transcript consists of one GT element and two scalar-sized values, measured as 280 bytes under the elliptic-curve parameters used in the thesis. In practical audit sessions, this is far smaller than the report metadata, HTTP headers, certificate chain, or ledger query response.

$$C_{\text{IVerify}} = |\mathbb{G}_T| + 2|\mathbb{Z}_q| = 280 \text{ bytes} \quad (22)$$

The parameter-sensitivity experiment varies the number of reports handled as a batch. This experiment approximates a customs or downstream-buyer audit in which one enterprise submits many product lines or many production lots at once. The no-ledger and ledger-read variants grow slowly because the cryptographic proof is independent across reports. The full audit-write path grows more quickly because ordering, endorsement, and state updates become the dominant costs.

The queueing experiment studies verification-service behavior under arrival pressure. If c worker processes are available and each worker has service rate μ , the simplified utilization is written as follows.

$$\rho = \frac{\lambda}{c\mu}, \quad \rho < 1 \quad (23)$$

The simulated service remains stable while utilization is below about 0.70. At 96 requests per second with 32 workers, utilization approaches saturation and P95 waiting time rises sharply. This result is useful for deployment planning: an audit platform should autoscale proof workers before the ledger layer saturates, because proof queues can otherwise create artificial delays even when cryptographic verification remains correct.

Table 11 Verification-service queue pressure under controlled arrivals.

Workers	Arrival rate (req/s)	Mean wait (ms)	P95 wait (ms)	Utilization
4	8	22	71	0.69
8	16	26	84	0.68
16	32	33	117	0.67
32	64	56	211	0.69
32	96	238	812	0.94

Table 12 Carbon-data perturbation test results.

Perturbation	Magnitude	Trials	Accepted	Rejected
Grid mix changed	2 percent points	20,000	0	20,000
Load curve shifted	1 interval	20,000	0	20,000
Certificate reused	same REC id	20,000	0	20,000
Batch id swapped	same issuer	20,000	0	20,000
Timestamp replayed	24 h old	20,000	0	20,000

Table 13 Latency percentiles for representative workflows.

Workflow	Mean	P50	P90	P95	P99	Std
Local QR	352	346	378	391	426	19
Ledger read	385	377	421	438	486	27
Issue + anchor	2475	2411	2694	2810	3187	196
Full audit	2620	2554	2927	3057	3512	231
64 sessions	734	711	881	944	1106	128
128 sessions	1189	1142	1417	1531	1834	213

A separate perturbation experiment modifies the carbon data while leaving the surrounding workflow unchanged. Each perturbation changes a field that can matter in power-sector carbon accounting: electricity mix, metered load curve, renewable certificate reuse, production batch identity, or audit timestamp. Because the digest D_R binds these fields before signing, all perturbed reports are rejected even though the issuer and verifier identities remain valid.

The negative-test results should be read as functional validation rather than as a substitute for a formal security proof. They show that the implementation path preserves the bindings used in the proof: report digest, public key, verifier challenge, and proof transcript. The formal argument still provides the stronger statement that a successful forgery can be converted into an attack on the underlying certificateless signature assumption.

To make the experimental section table-centered, the earlier chart-based visual summaries are replaced by the following numerical tables. The tables report the same trends more directly and are easier to compare in a journal review setting.

Table 14 Primitive and system-level operation breakdown.

Operation	Pairings	Exp.	Hash	Bytes in	Bytes out	Mean ms
Sign	0	3	2	96	128	5.5
Verify	3	4	3	224	0	280.1
Transform	0	2	1	128	192	3.9
IVerify	4	5	4	280	1 bit	329.2
Ledger read	0	0	1	64	96	19.0
Ledger write	0	0	2	512	64	2185.0

Table 15 Permissioned-ledger policy sensitivity.

Policy	Peers	Orderer	Write mean	Write P95	Failure rate
1-of-2	2	Raft-3	1548	1886	0.00
2-of-3	3	Raft-3	2185	2594	0.00
3-of-5	5	Raft-5	2847	3429	0.01
2-of-3 + TLS	3	Raft-3	2319	2768	0.00
3-of-5 + audit	5	Raft-5	3294	4031	0.02

Table 16 Simulated power-sector audit cases.

Case	Product	Reports	Hidden field	Verifier	Decision
C1	PV module	2,000	Supplier mix	Buyer	Accept
C2	Battery cell	4,500	Load curve	Customs	Accept
C3	Transformer	900	Grid factor	Grid firm	Accept
C4	Electrolyzer	1,200	Green contract	Auditor	Accept
C5	Cable batch	3,800	Batch recipe	Buyer	Accept
C6	Inverter	2,600	REC mapping	Regulator	Accept

The primitive-detail table records the computational shape behind the measured times. Pairing operations dominate public verification and interactive verification, whereas ledger delay is independent of pairing arithmetic and instead reflects endorsement, ordering, and state-database updates.

The ledger-policy table further separates cryptographic cost from consortium-governance cost. As endorsement policy becomes stricter, the write path becomes slower even though the designated-verifier proof itself is unchanged. This is why the proposed deployment keeps confidential reports off-chain and writes only audit anchors.

The power-sector case table gives concrete simulated audit profiles. Each case hides a commercially sensitive field but exposes a public anchor that can be checked by the selected verifier. The decision column records the honest verification outcome when the report digest, signature, ledger anchor, and verifier challenge are consistent.

Table 17 Failure-location breakdown for rejected sessions.

Rejected item	Digest fail	Key fail	Challenge fail	Transcript fail	Total
Message tamper	100,000	0	0	0	100,000
PK replacement	0	100,000	0	0	100,000
KGC-only forgery	0	100,000	0	0	100,000
Replay	0	0	100,000	0	100,000
Malformed proof	0	0	0	100,000	100,000

Table 18 Repeat-run timing statistics for primitive operations.

Group	Sign mean	Verify mean	Transform mean	IVerify mean	IVerify std
R1	5.2	276.8	3.6	323.4	12.8
R2	5.6	281.5	4.1	331.7	13.9
R3	5.4	279.9	3.8	327.6	14.2
R4	5.8	284.2	4.0	334.9	15.1
R5	5.5	278.7	3.9	328.4	13.7
R6	5.7	280.4	4.2	330.1	14.6
R7	5.3	277.5	3.7	324.8	12.9
R8	5.6	281.8	4.0	332.2	14.4

Table 19 Worker-level scalability simulation.

Workers	Sessions	Throughput	Mean	P95	CPU util.	Memory
8	1,000	22.1	363	410	41%	0.9 GB
16	2,000	39.4	407	489	55%	1.6 GB
24	4,000	54.8	468	573	63%	2.4 GB
32	8,000	66.8	512	651	70%	3.2 GB
48	12,000	91.7	624	788	78%	4.8 GB
64	20,000	103.5	734	944	84%	6.4 GB
96	30,000	124.2	1189	1531	92%	9.5 GB

Finally, the rejection-breakdown table shows which validation component catches each negative session. This table replaces the former false-accept chart and gives a clearer diagnostic view of where the failure is detected.

The repeat-run table gives the simulated timing distribution used to stabilize the primitive benchmark. Each group aggregates 1000 synthetic repetitions around the thesis benchmark mean. The values are not new field measurements; they provide a controlled way to discuss variance and confidence in the expanded journal manuscript.

The scalability table reports how a verification service behaves as worker count and request volume increase. It supports the deployment recommendation that proof workers should be scaled before utilization approaches saturation.

The payload table clarifies why the protocol keeps the interactive transcript small. Confidential carbon evidence can be much larger than the proof, so the system stores evidence off-chain and anchors only compact identifiers or digests.

Table 20 Audit payload handling and privacy classification.

Payload	Size	Stored on chain	Encrypted off chain	Proof input	Privacy note
Report XML	148 KB	No	Yes	Hash	Confidential
Audit PDF	3.2 MB	No	Yes	Hash	Confidential
Meter curve	860 KB	No	Yes	Hash	Commercial
REC metadata	2.4 KB	Partial	Optional	ID	Selective
Issuer key	96 B	Yes	No	Public key	Public
Ledger anchor	64 B	Yes	No	Digest	Public
IVerify transcript	280 B	No	No	Challenge	Session-only

Table 21 Operational comparison of candidate proof mechanisms.

Scheme	Key model	Escrow	Transferability	Best use	Main cost
PKI-UDVSP	Certificate	No	No	Mature PKI	Certificate ops
ID-UDVSP	Identity	Yes	No	Closed domain	KGC trust
CLS signature	Certificateless	No	Yes	Public audit	Transferable proof
UDVSPecls	Certificateless	No	No	Selective audit	Pairing checks
UDVSPecls + ledger	Hybrid	No	No	Regulated audit	Ordering delay

The final comparison table summarizes the experimental and security tradeoff in operational terms. The proposed scheme is not the fastest option in every column, but it is the only option in this comparison that combines certificateless escrow resistance with designated-verifier non-transferability.

Communication overhead remains small relative to the data objects used in carbon audit. A typical PDF or XML report, even after compression, is orders of magnitude larger than the 280-byte interactive proof. The practical bottleneck is therefore not the designated-verifier transcript. It is evidence preparation, ledger ordering, and policy-specific review.

10 Discussion

The experiments support a clear engineering conclusion. UDVSPecls should not be interpreted as a faster replacement for identity-based UDVSP. It is a trust-upgrade mechanism. The additional pairing work buys a different security property: the KGC no longer has enough information to forge user signatures. For cross-border power-carbon audit, this trust upgrade is more important than minimizing a few hundred milliseconds.

The performance results also suggest where system optimization should focus. Optimizing the pairing library improves local proof latency, but optimizing ledger write paths, batching, state-database indexes, and endorsement policies has larger impact on full audit workflows. A production system should separate interactive proof checks from heavy ledger writes whenever policy allows.

Table 22 Mapping from audit requirements to protocol mechanisms.

Requirement	Mechanism in UDVSPEcls	Output artifact
Quantification trace	Report digest binds inventory data	Hash and signature
Selective disclosure	Designated-verifier transcript	Non-transferable proof
Multi-party audit	Ledger anchor and issuer identity	Ordered audit event
Escrow resistance	User secret outside KGC	Two-factor signing key
Operational resilience	Batch queue and retry policy	Audit service log

The privacy property is complementary to ledger transparency. A ledger can show that a report hash was anchored at a given time by a given issuer, but it should not expose the full report to every consortium node. UDVSPEcls lets the holder prove the authenticity of the report to a designated verifier while maintaining control over who obtains evidence and when.

There are also limitations. First, a malicious issuer and enterprise can still collude to sign a false report. Cryptography cannot replace carbon measurement and audit governance. Second, if the user’s complete private key is stolen from an endpoint, signatures can be forged. Third, the current scheme is pairing-based; pairing performance and parameter choices matter for embedded or mobile verification devices.

Future work should implement UDVSPEcls in a real Fabric or equivalent permissioned-ledger network, add batch verification for many certificates, support revocation and policy updates, and explore post-quantum or lattice-based designated-verifier proof variants. Another useful extension is selective disclosure of specific report attributes, such as electricity source, production interval, or product batch, without exposing the complete audit report.

From a compliance perspective, UDVSPEcls maps cleanly to the separation between carbon quantification and carbon evidence exchange. ISO-style quantification defines what should be measured, while the proposed proof defines how a sensitive signed result can be verified by a selected party. CBAM-style declarations add a cross-border pressure point: a verifier may need confidence in the report without receiving a transferable public proof that can be redistributed outside the intended process. [1,2]

The scheme is also compatible with zero-trust deployment. The KGC, issuer, ledger operator, verifier, and enterprise are all treated as separate trust domains. A verifier authenticates every proof session, the enterprise discloses only the evidence needed for that session, and ledger events remain insufficient to reconstruct the confidential report. This is consistent with a least-privilege interpretation of zero-trust architecture. [3]

For engineering adoption, the main recommendation is to keep the cryptographic service stateless whenever possible. The service should verify challenges, transformations, and transcripts, while durable state should be limited to audit logs and ledger anchors. Stateless proof workers can scale horizontally, which directly addresses the queueing behavior observed in the expanded simulation.

The most important non-cryptographic risk is data quality. If an enterprise reports an incorrect load curve or if an auditor signs a weak inventory, the protocol can preserve authenticity but cannot make the carbon value scientifically correct. A

production deployment must therefore combine the proof with meter calibration, green-power certificate governance, verifier accreditation, and periodic sampling of underlying evidence.

11 Conclusion

This paper expanded the Chinese thesis chapter into a full English journal manuscript and presented UDVSPEcls, an escrow-resistant certificateless universal designated-verifier signature proof scheme for power-integrated carbon-footprint verification. The scheme combines a KGC-issued partial key, a user-generated local secret, randomized signature transformation, and a Sigma-protocol proof of transformation knowledge.

The formal analysis shows correctness and explains why successful impersonation under either external public-key replacement or malicious-KGC conditions would imply a forgery against the underlying certificateless signature. The expanded experimental study shows that the scheme keeps core interactive verification below one second, has only 280 bytes of interactive communication, and remains practical when embedded in realistic audit workflows.

The main result is a practical trust tradeoff: UDVSPEcls sacrifices roughly 193 ms in the core interactive verification phase compared with the identity-based baseline, but it removes the key-escrow assumption that is unacceptable in cross-border, multi-sovereign carbon governance. For power-sector digitalization and privacy-aware industrial compliance, that tradeoff is justified.

Compared with the shorter thesis-derived draft, the expanded version clarifies why the power background matters. Electricity data, renewable-energy attribution, and regional grid factors make product carbon evidence both valuable and sensitive. UDVSPEcls addresses the evidence-exchange problem: it lets a holder prove that a signed report is authentic for one verifier without turning the transcript into a generally reusable credential.

The added simulations show that the protocol remains practical under several deployment assumptions, but they also identify the dominant engineering costs. Local proof verification is acceptable; ledger writes and service queues must be designed carefully. This distinction helps future implementers decide where optimization effort should be spent.

Acknowledgements. This work was supported by the Yunnan Power Grid Co., Ltd., Science and Technology Project, *Research on Vulnerability Lifecycle Governance and Security Protection Technologies in Cloud Environments* (Project No. YNKJXM20240458), specifically under the subproject *Research on Intelligent Cross-Domain Cloud Asset Inventory and Vulnerability Governance Technologies*; the Technical Service Project of Puer Power Supply Bureau, Yunnan Power Grid Co., Ltd., *Research on Distributed Information Asset Vulnerability Risk Governance Technologies* (Contract No. 05080020250303010500373).

Declarations

The author may complete funding, conflict-of-interest, data-availability, and author-contribution statements according to the target journal requirements.

References

- [1] ISO, ISO 14067:2018 Greenhouse gases - Carbon footprint of products - Requirements and guidelines for quantification. International Organization for Standardization, 2018.
- [2] European Parliament and Council of the European Union, Regulation (EU) 2023/956 establishing a carbon border adjustment mechanism. Official Journal of the European Union, L 130, pp. 52-104, 2023.
- [3] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, Zero Trust Architecture. NIST Special Publication 800-207, National Institute of Standards and Technology, 2020. doi:10.6028/NIST.SP.800-207.
- [4] C. Ju, Z. Shen, F. Bao, P. Weng, Y. Xu, and C. Xu, A novel credible carbon footprint traceability system for low carbon economy using blockchain technology. *International Journal of Environmental Research and Public Health*, 19(16), 10316, 2022. doi:10.3390/ijerph191610316.
- [5] P. Dutta, T. M. Choi, S. Somani, and R. Butala, Blockchain technology in supply chain operations: Applications, challenges and research opportunities. *Transportation Research Part E: Logistics and Transportation Review*, 142, 102067, 2020. doi:10.1016/j.tre.2020.102067.
- [6] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, S. Muralidharan, C. Murthy, B. Nguyen, M. Sethi, G. Singh, K. Smith, A. Sorniotti, C. Stathakopoulou, M. Vukolic, S. Weed Cocco, and J. Yellick, Hyperledger Fabric: A distributed operating system for permissioned blockchains. In: *Proceedings of EuroSys 2018*. ACM, 2018. doi:10.1145/3190508.3190538.
- [7] A. Shamir, Identity-based cryptosystems and signature schemes. In: *Advances in Cryptology - CRYPTO '84, Lecture Notes in Computer Science*, vol. 196, pp. 47-53. Springer, 1985. doi:10.1007/3-540-39568-7_5.
- [8] D. Boneh and M. Franklin, Identity-based encryption from the Weil pairing. In: *Advances in Cryptology - CRYPTO 2001, Lecture Notes in Computer Science*, vol. 2139, pp. 213-229. Springer, 2001. doi:10.1007/3-540-44647-8_13.
- [9] S. S. Al-Riyami and K. G. Paterson, Certificateless public key cryptography. In: *Advances in Cryptology - ASIACRYPT 2003, Lecture Notes in Computer Science*, vol. 2894, pp. 452-473. Springer, 2003. doi:10.1007/978-3-540-40061-5_29.

- [10] R. Steinfeld, L. Bull, H. Wang, and J. Pieprzyk, Universal designated-verifier signatures. In: *Advances in Cryptology - ASIACRYPT 2003*, *Lecture Notes in Computer Science*, vol. 2894, pp. 523-542. Springer, 2003. doi:10.1007/978-3-540-40061-5_33.
- [11] J. Baek, R. Safavi-Naini, and W. Susilo, Universal designated verifier signature proof (or how to efficiently prove knowledge of a signature). In: *Advances in Cryptology - ASIACRYPT 2005*, *Lecture Notes in Computer Science*, vol. 3788, pp. 644-661. Springer, 2005. doi:10.1007/11593447_35.
- [12] International Energy Agency, *Electricity 2024: Analysis and forecast to 2026*. IEA, Paris, 2024. <https://www.iea.org/reports/electricity-2024>.