

DOI: 10.37943/RPCC6318

Tolegen Aidynov

PhD Student, Department of Information Security
tolegen.ch@gmail.com; orcid.org/0009-0005-3327-9719
L.N. Gumilyov Eurasian National University, Kazakhstan

Dina Satybaldina

Candidate of Physical and Mathematical Sciences, Professor
Head of the Research Institute of Information Security and Cryptology
satybaldina_dzh@enu.kz; orcid.org/0000-0003-0291-4685
L.N. Gumilyov Eurasian National University, Kazakhstan

Willian Dimitrov

PhD, Professor, Department of Computer Science
v.dimitrov@unibit.bg; orcid.org/0000-0002-5348-4948
University of Library Studies and Information Technologies, Bulgaria

**VOTINGONTOSEC: A SECURITY ONTOLOGY FOR RISK MODELING IN
ELECTRONIC VOTING SYSTEMS**

Abstract: Blockchain voting needs a security model for shared infrastructure, cryptographic tools, and election rules. General ontologies do not jointly cover its threats and standards-based risk assessment. This paper presents VotingOntoSec, an OWL ontology aligned with ISO/IEC 27005. It represents assets, threats, vulnerabilities, risks, and countermeasures, including Sybil and smart-contract attacks and cryptographic protections. Development in Protégé 5.6.1 was followed by graph-based structural tests, HermiT 1.4.3 reasoning, competency-question assessment, comparison with five representative ontologies, and a Smart City voting scenario. The ontology contains 118 classes, 16 object properties, and 9 data properties. Its class hierarchy has no cycles, and HermiT found all named classes satisfiable. A five-asset example illustrates traceability from a vulnerable asset through an exploiting threat to a suitable control. Expert-assigned likelihood and criticality estimates yield reproducible inherent and residual risk scores. In the illustrative calculation, the criticality-weighted system score decreases from 0.81 to 0.32 after applying modeled controls; rankings remain unchanged as the likelihood weight varies from 0.3 to 0.7. Among the compared models, VotingOntoSec uniquely combines voting-specific blockchain threats, cryptographic countermeasures, and ISO/IEC 27005-aligned risk analysis. The case includes voter data, the ledger, consensus nodes, a communication channel, and a voting device. Results help rank needed controls and show how risk may change as inputs or weights change. The numerical example is illustrative rather than an empirical performance evaluation. The ontology and scripts are available for independent verification and reuse. VotingOntoSec provides a new practical semantic basis for structured security assessment of decentralized voting systems at scale.

Keywords: blockchain electronic voting; security ontology; risk modeling; ISO/IEC 27005; Smart City; cryptographic protocols; OWL.

Introduction

Electronic voting may make elections easier to access and more transparent. It still has to protect voter privacy, preserve ballot integrity, and let people verify the results [1]. Blockchain designs, combined with cryptographic methods, can safeguard ballot records and make election processes independently verifiable [2–3]. A proper assessment must connect assets with threats, vulnerabilities, and controls, following the approach set out in ISO/IEC 27005:2022 [4]. VotingOntoSec maps those relationships for blockchain-based voting and supports semantic queries as well as illustrative risk calculations.

The study tests whether a clearly defined voting-security ontology can link assets, threats, vulnerabilities, and controls without confusing property directions. It also checks the model for structural and logical consistency, then applies it to risk assessments involving several assets. The result is a semantic model accompanied by a transparent scoring example—not evidence that blockchain voting is inherently secure.

Reviews point to blockchain voting’s potential while also documenting its continuing security problems [5–6]. Adding biometric checks or post-quantum methods creates more architectural decisions [7–8]. Smart City platforms bring together varied services and analytics [9–10], so assessing how a governance-critical voting system works across those platforms has clear practical value.

Security ontologies have been applied to assurance metrics, standards-based control models [11], AI security, and enterprise architecture [12–15]. Research on social-engineering knowledge graphs and ontology quality also helps shape domain models [16–17], while information-security governance studies support explicit treatment of risk [18–19]. Surveys of smart-contract verification point to another gap: blockchain-specific threats must be represented as well [20]. None of the models compared here brings together voting assets, blockchain and cryptographic threats, countermeasures, and ISO/IEC 27005-aligned scoring. We assess this combination using structural and logical checks, comparative analysis, and a worked voting scenario.

Methods and Materials

VotingOntoSec was built in OWL 2 from ISO/IEC 27005:2022 concepts and published voting-security literature. We specified competency questions, defined the five risk concepts and their directional relations, populated voting-specific instances, and corrected inconsistencies during review. The ontology covers five key concepts and their interrelationships (Fig. 1).

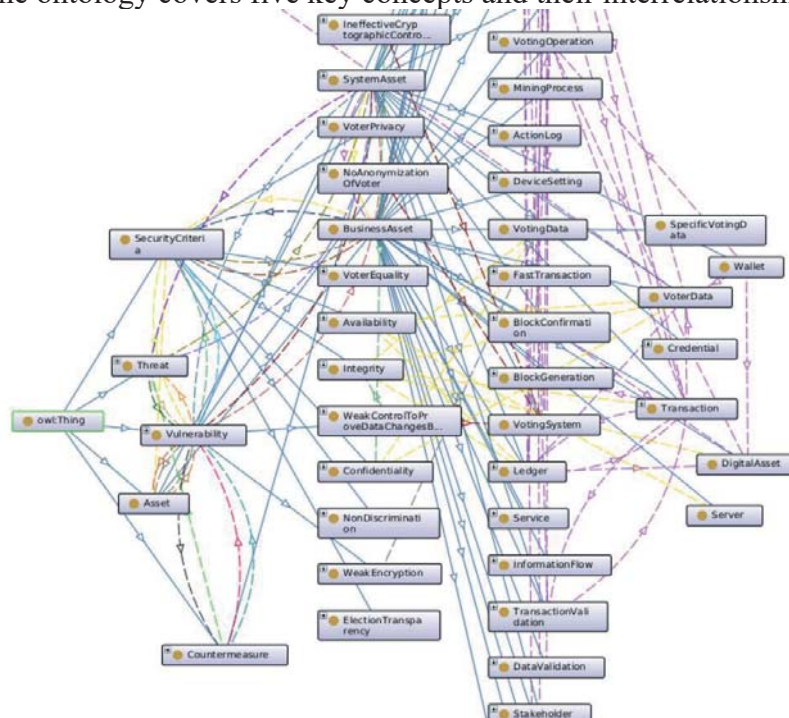


Figure 1. Main concepts and their interrelationships in VotingOntoSec

The principal relations are Asset isHarmedBy Vulnerability, Threat exploits Vulnerability, Threat affects Asset, Countermeasure mitigates Threat, and Countermeasure reduces Vulnerability. The inverse properties, exploitedBy and mitigatedBy, point opposite to the

directions stated by their names. Risk connects the resulting threat, vulnerability, and asset exposure. Security criteria sit in a separate branch that cuts across the ontology.

We edited the ontology in Protégé 5.6.1, used a Python 3.10 script to inspect its hierarchy, and classified it with HermiT 1.4.3. Structural checks examined the roots, leaves, hierarchy depth, subclass edges, property directions, and possible cycles. Logical checks focused on consistency and satisfiability. The numerical case relies on expert-assigned inputs to show how the calculations work; these values are not measured incident probabilities. Three checks supported the review. A graph traversal counted classes, roots, leaves, subclass edges, and hierarchy depth while also searching for cycles.

We manually compared each object property's name with its domain, range, and inverse. HermiT classified the OWL ontology and checked its global consistency, class satisfiability, and inferred subclass relationships. These tests describe the model's properties; they do not, by themselves, confirm the experts' risk ratings or prove that the controls work in practice. The threat taxonomy covers Sybil, eclipse, and consensus attacks, along with data theft, deanonymization, replay, endpoint compromise, and smart-contract exploitation. Each threat and control stays tied to the assets and vulnerabilities it affects.

The model uses five core concepts: Asset, Threat, Vulnerability, Risk, and Countermeasure. Security criteria-including privacy, integrity, availability, equality, and transparency-define what the controls are meant to protect. For instance, voter data is an Asset harmed by weak anonymity. A deanonymization Threat exploits that Vulnerability and affects the Asset, while a homomorphic-encryption Countermeasure limits the threat or reduces exposure. Risk captures that specific combination of threat, vulnerability, and asset. The passive relation exploitedBy runs from the Vulnerability to the Threat, while the inverse exploits runs from the Threat to the Vulnerability. This explicit direction allows the competency questions to be answered consistently.

Results

The ontology defines 118 classes, 16 distinct object properties, 9 data properties, and 671 axioms (Table 1). The 29 class-level relationship axioms differ from the 16 distinct property types and the 112 subclass links.

The count of 16 object properties refers to relation types, whereas 29 denotes class-level relationship axioms and 112 denotes subclass links. The taxonomy is intentionally shallow. Absence of multiple inheritance describes its design and should not be mistaken for a general requirement of OWL; absence of subclass cycles is the relevant structural check.

Table 1 – Basic metrics of the VotingOntoSec ontology

Metric	Count
Axioms	671
Logical axioms	265
Classes	118
Object properties	16
Data properties	9
Root classes	6
Leaf classes	110
Maximum hierarchy depth	3
Subclass cycles	0
Multiple inheritance	0

Competency questions test whether the ontology can retrieve exposed assets, their vulnerabilities and threats, associated risks, and available countermeasures (Table 2). Property domain and range directions were checked against these queries.

Table 2 – Competency questions answered by VotingOntoSec

CQ	Question	Answered through (ontology construct / example)
CQ1	Which threats affect a given asset?	Threat affects Asset (inverse of isHarmedBy); e.g., DeanonymizationAttack affects VoterData
CQ2	Which vulnerabilities does a given threat exploit?	Threat exploits Vulnerability (the inverse of exploitedBy, whose direction is Vulnerability exploitedBy Threat); e.g., DeanonymizationAttack exploits LackOfAnonymity
CQ3	Which vulnerabilities are present on a given asset?	Asset isHarmedBy Vulnerability; e.g., VoterData isHarmedBy LackOfAnonymity
CQ4	Which countermeasures mitigate a given threat or reduce a given vulnerability?	Threat mitigatedBy Countermeasure (inverse: Countermeasure mitigates Threat); Countermeasure reduces Vulnerability; e.g., HomomorphicEncryption mitigates DeanonymizationAttack
CQ5	Which risks are associated with an asset, and what are their impact and likelihood?	Asset hasRisk Risk; each Risk carries hasImpactLevel and hasLikelihood (data properties)
CQ6	Which assets remain unprotected (no countermeasure for their vulnerabilities)?	Query for an Asset with a Vulnerability but no linked mitigating Countermeasure
CQ7	Which blockchain-specific threats apply to consensus and ledger assets?	Subclasses of Threat (SybilAttack, 51PercentAttack, SmartContractThreat) linked to consensus/ledger assets
CQ8	Which security criteria does a given countermeasure support?	Countermeasure related to SecurityCriteria (confidentiality, integrity, availability, voter privacy, ...)
<i>Note - compiled by the authors</i>		

The graph has six roots, 110 leaves, a maximum depth of three, no subclass cycles, and no multiple inheritance. The latter is a deliberate taxonomy choice, not an OWL correctness requirement. Hermit found a globally consistent ontology in which all 118 named classes are satisfiable (Table 3).

The six root classes and 110 leaves produce a broad hierarchy with eight internal classes. Its maximum depth is three, so a new voting attack or control can normally be added as a leaf under an established category. Graph metrics characterize maintainability, while the reasoner result in Table 3 addresses logical satisfiability. Neither test alone establishes that every possible attack has been captured.

Table 3 – Reasoner-based logical verification

Verification item	Result
Reasoner (version)	HermiT 1.4.3 (bundled in Protege 5.6.1)
OWL profile	OWL 2 DL
Global consistency	Consistent – no logical contradictions
Unsatisfiable classes	0 – all named classes satisfiable

We compared VotingOntoSec with OVM, CS-CASEOnto, and three published ontologies covering ISO 27002 controls, ArchiMate security, and AI-enabled systems [13-15]. The selection was purposive, so the comparison establishes distinctiveness within this set rather than across all ontologies. Only VotingOntoSec combines the blockchain-voting domain, explicit cryptographic mechanisms, and ISO/IEC 27005-aligned risk scoring in this set (Table 4).

The five comparison models were chosen from prior work with documented security concepts and distinct target domains. The dimensions in Table 4 record only support evidenced by their published descriptions. A 'partial' rating denotes incomplete or indirect coverage. The comparison is qualitative and bounded by the selected set; it should not be read as a systematic review of every security ontology.

Table 4 – Comparison with representative security ontologies

Dimension	OVM	CS-CASEOnto	Fenz ISO 27002 [15]	ArchiMate sec. [14]	AI-CS [13]	VotingOntoSec
Domain focus	Vuln. mgmt	Cloud / enterprise	ISO 27002 controls	Enterprise architecture	AI-enabled systems	Blockchain e-voting
Blockchain-specific threats	No	No	No	No	No	Yes
Cryptographic mechanisms	No	Partial	No	Partial	No	Yes
ISO/IEC 27005 alignment	Partial	Yes	No (27002)	No	No	Yes
Risk quantification	No	Partial	Partial	No	Partial	Yes (formal model, heat map)
Formal reasoning (DL)	No	Partial	Yes	Yes	Yes	Yes (HermiT)
Public availability	Partial	No	Yes	Yes	Partial	Yes
<i>Note – compiled by the authors</i>						

For the illustrative five-risk case, an expert assigns likelihood P and asset criticality C on $[0, 1]$. We use the transparent weighted score $R_{inh} = wP + (1 - w)C$, with $w = 0.5$ by default; ISO/IEC 27005 does not prescribe this arithmetic rule. Table 5 records the underlying scores and exposes their assumptions.

To make each input interpretable, a criticality score reflects the likely consequence for an asset, while likelihood denotes the estimated chance of the paired threat. Equal weights provide a neutral default. The user of the ontology may select another weight according to local priorities, and the sensitivity check below shows how much the five case scores move under that choice.

Higher scores denote greater treatment priority rather than a statistically calibrated attack probability.

The estimates are illustrative rather than observed probabilities. The mapping from qualitative bands to representative midpoints (very low through very high: 0.1, 0.3, 0.5, 0.7, 0.9) permits local reassessment without changing the ontology.

Table 5 - Threat likelihood, asset criticality and computed risk level per vulnerability

Vulnerability	Threat likelihood P	Asset criticality C	Risk level R = (P+C)/2
Insecure communication control	0.80	1.00	0.90
Fake identity creation	0.75	0.95	0.85
Weak authentication	0.70	0.90	0.80
Ineffective cryptographic control	0.65	0.85	0.75
Lack of anonymization of voter	0.60	0.80	0.70
<i>Note – compiled by the authors</i>			

The municipal-election example links five assets-voter data, smart contract/ledger, consensus nodes, communication channel, and voting device-to vulnerabilities, exploiting threats, and controls (Table 6; Fig. 2). Each risk refers to a threat exploiting a vulnerability against a particular asset; it is not an intrinsic attribute of a vulnerability alone.

The qualitative labels in Table 6 describe a separate trace through the five assets; they are not rounded versions of the numerical inputs in Table 5. Only the values in Table 5 underpin the illustrative calculations below.

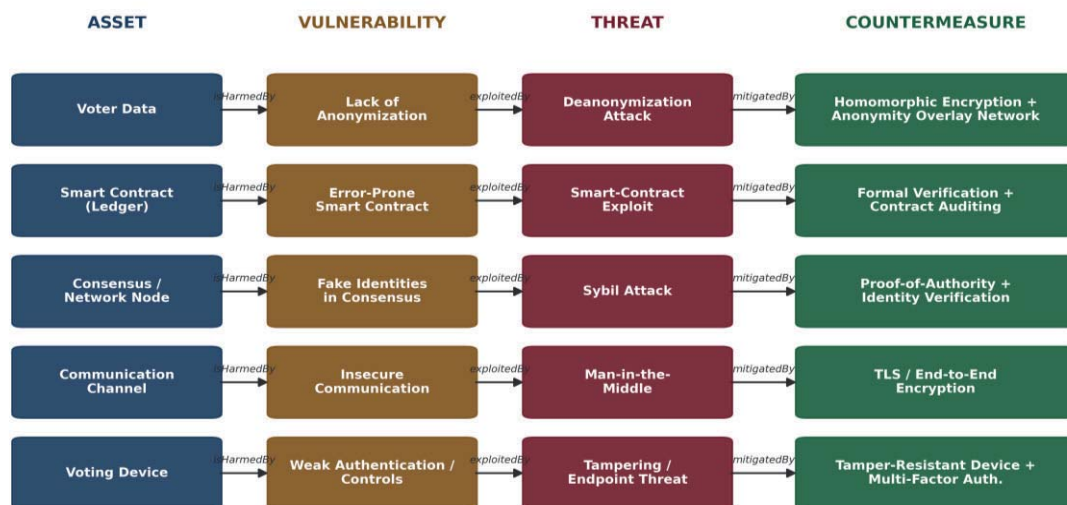


Figure 2. Five-asset risk-modeling scenario in VotingOntoSec

Equations (1)-(2) can be represented in a compact formal model so that the risk calculation is reproducible and each quantity maps directly to an ontology construct. Let A, T, V, and M represent the sets of assets, threats, vulnerabilities, and countermeasures instantiated in VotingOntoSec. A risk is therefore a valid threat-vulnerability-asset triple:

$$R = \{(t, v, a) \in T \times V \times A : \text{exploits}(t, v) \wedge \text{isHarmedBy}(a, v) \wedge \text{affects}(t, a)\} \quad (1)$$

so that the Risk entity of the ontology is the reification of an element $\rho = (t, v, a) \in \mathcal{R}$ - exactly the “per-vulnerability” reading defined above, in which a risk is the exposure created when threat t exploits vulnerability v on asset a .

Every threat is assigned a qualitative likelihood, $\lambda(t)$, while each asset receives a qualitative criticality rating, $\sigma(a)$. Both use the five ISO/IEC 27005 bands-VL, L, M, H, VH}-which two

monotone mapping functions then translate into normalized values between 0 and 1 for computation:

$$P = \ell(\lambda(t)), \quad C = \kappa(\sigma(a)), \quad \ell, \kappa : \{VL, L, M, H, VH\} \rightarrow [0, 1] \quad (2)$$

with representative midpoints $\ell = \kappa : (VL, L, M, H, VH) \mapsto (0.1, 0.3, 0.5, 0.7, 0.9)$; as noted for Table 6, an expert may refine a value within its band. The inherent (pre-treatment) risk of ρ is the weighted mean already given in Equation (3), $R_{inh}(\rho) = w_1P + w_2C$, which reduces to Equation (1) when $w_1 = w_2 = 0.5$.

The ontology connects each vulnerability with the countermeasures that address it: mitigates $\subseteq M \times T$ and reduces $\subseteq M \times V$, as specified in competency question CQ4. Each countermeasure μ receives an effectiveness score $\eta(\mu) \in [0, 1]$. If independent controls target the same factor, their effects combine according to the probabilistic-OR rule:

$$\eta = 1 - \prod(1 - \eta_k), \quad k = 1, \dots, q \quad (3)$$

A control may lower the likelihood of successful exploitation, the consequence, or both, giving the residual likelihood and impact factors:

$$P_{res} = P(1 - \eta_P), \quad C_{res} = C(1 - \eta_C) \quad (4)$$

and hence the residual risk and its relative reduction:

$$R_{res}(\rho) = w_1P_{res} + w_2C_{res}, \quad \delta(\rho) = (R_{inh} - R_{res}) / R_{inh} \quad (5)$$

Table 6 evaluates Equations (3)-(5) for the same five risks, using the countermeasures already modeled for each asset, and equal weights $w_1 = w_2 = 0.5$.

Table 6 - Residual risk after applying the modeled countermeasures ($w_1 = w_2 = 0.5$, $\tau = 0.35$)

Vulnerability	Countermeasure(s)	η_P	η_C	R_{inh}	R_{res}	δ (%)	Treat?
Insecure communication control	TLS / end-to-end encryption	0.80	0.50	0.90	0.33	63.3	No
Fake identity creation	Proof-of-authority; identity verification	0.88	0.20	0.85	0.43	50.0	Yes
Weak authentication	Multi-factor auth.; tamper-resistant device	0.85	0.30	0.80	0.37	54.1	Yes
Ineffective cryptographic control	Post-quantum / strong cryptography	0.75	0.60	0.75	0.25	66.5	No
Lack of anonymization of voter	Homomorphic encryption; anonymity overlay	0.80	0.70	0.70	0.18	74.3	No
Note – compiled by the authors							

Two findings matter most. Treatment cuts each risk by 50–74%, giving a numerical measure of the protection represented in the ontology. The ranking shifts as well: insecure communication starts as the greatest inherent risk but drops to third after strong transport and end-to-end encryption are introduced. Fake-identity creation-the Sybil vector-then ranks first among the remaining risks. Identity controls make it less likely, but the consensus layer remains highly critical.

Individual risks can be combined into measures for an asset or an entire system. If the risks affecting one asset are treated as independent, their combined value is calculated using the probabilistic-OR formula:

$$R_{\text{asset}}(a) = 1 - \prod (1 - R(\rho)), \quad \rho \in \mathcal{R}(a) \quad (6)$$

which is a conservative upper bound, since it saturates toward 1 when many risks share one asset. For reporting we therefore prefer the criticality-weighted system risk:

$$R_{\text{sys}} = (\sum C(a) R_{\text{asset}}(a)) / (\sum C(a)), \quad a \in A \quad (7)$$

Using asset criticality as the weight, Equation (7) gives the five risks in Table 6 an inherent system score of 0.81, placing them in the very-high band.

Discussion

VotingOntoSec links formal ontology queries to a reproducible, standards-aligned risk-assessment example. The class and reasoner checks establish structural and logical properties; the case and comparison demonstrate intended use within the tested scope. The scores depend on expert inputs, while the comparison includes only five selected alternatives.

The ontology has not been tested in a live voting deployment. It does not ingest real-time threat intelligence, and its risk ratings and control effectiveness require independent expert elicitation and empirical validation.

The chief practical benefit is traceability: analysts can inspect why a risk is associated with an asset and which modeled controls address it. The competency questions establish query coverage, and the five-asset example shows that the same relation scheme can represent different attack paths. However, a successful ontology query does not prove that the underlying threat catalogue is complete, that a proposed control will be deployed correctly, or that its estimated effectiveness will generalize.

Future work will evaluate the model with election-security experts, test it against deployed systems, and add dynamic evidence feeds.

Conclusion

This research proposed VotingOntoSec, a domain-specific security ontology for blockchain-based electronic voting systems. The ontology standardizes the core information-security concepts - assets, threats, vulnerabilities, risks, and countermeasures - and aligns its structure with ISO/IEC 27005 risk treatment to support systematic threat modeling and risk analysis. Its distinctiveness lies in combining blockchain-specific attacks (e.g., Sybil attacks and smart-contract vulnerabilities), cryptographic countermeasures, and formal risk-modeling principles within a single, consistent semantic framework.

Topological and conceptual analysis confirmed that the ontology - comprising 118 classes, 16 object properties, and 9 data properties - is structurally sound, consistent, and free of cyclic dependencies and multiple-inheritance conflicts, and is therefore suitable for further development. Logical verification using the HermiT reasoner confirmed that the ontology is globally consistent and that every named class is satisfiable. On that basis, it is logically validated and suitable for further development. A simplified use case demonstrated how the key security components of a blockchain-based voting system can be modeled and assessed, and comparison with existing ontologies confirmed the added value of VotingOntoSec for secure decentralized voting.

The work responds to the absence of ontologies that model blockchain-based voting risks in accordance with international standards; because no ready-made structures addressed both blockchain-specific threats and ISO-compliant risk frameworks, the ontology was built from the ground up. Future work will expand VotingOntoSec to incorporate dynamically acquired threat intelligence and real-world risk adaptation, and will extend it to related domains such as decentralized identity management and blockchain-based supply-chain security. In addressing these problems, VotingOntoSec provides a foundation for formal, analyzable, and interoperable electronic voting and other decentralized systems.

Data availability and reproducibility

To allow the results to be checked independently, we provide the complete VotingOntoSec ontology as an OWL 2 file in RDF/XML serialization, along with the Python scripts for the topological analysis and heat-map computation. The ontology, its versioned releases, a machine-

readable license, and step-by-step instructions for reproducing the reasoning and metrics are available in a public repository: <https://github.com/Tolegen95/VotingOntoSec> (the version described in this paper corresponds to release v1.0 of the repository, so that reviewers can retrieve the exact artefact). Readers and reviewers can open the file in Protege 5.6.1.

Funding: This research has been funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No. BR24992852, “Intelligent models and methods of Smart City digital ecosystem for sustainable development and the citizens’ quality of life improvement”).

References

- [1]. Darmawan, I. (2021). E-voting adoption in many countries: A literature review. *Asian Journal of Comparative Politics*, 6(4), 482–504. <https://doi.org/10.1177/20578911211040584>
- [2]. Jayakumari, B., Sheeba, S. L., Eapen, M., Anbarasi, J., Ravi, V., Suganya, A., & Jawahar, M. (2024). E-voting system using cloud-based hybrid blockchain technology. *Journal of Safety Science and Resilience*, 5(1), 102-109. <https://doi.org/10.1016/j.jnlssr.2024.01.002>
- [3]. Kiayias, A., Zacharias, T., & Zhang, B. (2015). End-to-end verifiable elections in the standard model. In E. Oswald & M. Fischlin (Eds.), *Advances in Cryptology – EUROCRYPT 2015* (pp. 468-498). Springer. https://doi.org/10.1007/978-3-662-46803-6_16
- [4]. International Organization for Standardization. (2022). *ISO/IEC 27005:2022 Information security, cybersecurity and privacy protection - Guidance on managing information security risks*. ISO. <https://www.iso.org/standard/80585.html>.
- [5]. Hajian Berenjestanaki, M., Barzegar, H. R., El Ioini, N., & Pahl, C. (2024). Blockchain-based e-voting systems: A technology review. *Electronics*, 13(1), 17. <https://doi.org/10.3390/electronics13010017>
- [6]. Barelli, R., D'Onghia, M., & Longari, S. (2025). Towards secure electronic voting: A survey on e-voting systems and attacks. *IEEE Access*, 13, 89600-89626. <https://doi.org/10.1109/ACCESS.2025.3569334>
- [7]. Ohize, H. O., Onumanyi, A. J., Umar, B. U., Ajao, L. A., Isah, R. O., Dogo, E. M., Nuhu, B. K., Olaniyi, O. M., Ambafi, J. G., Sheidu, V. B., & Ibrahim, M. M. (2025). Blockchain for securing electronic voting systems: A survey of architectures, trends, solutions, and challenges. *Cluster Computing*, 28(2), 132. <https://doi.org/10.1007/s10586-024-04709-8>
- [8]. Joni, S. A., Rahat, R., Tasnin, N., Ghose, P., Uddin, M. A., & Ayoade, J. (2024). Hybrid-blockchain-based electronic voting machine system embedded with Deepface, sharding, and post-quantum techniques. *Blockchains*, 2(4), 366-423. <https://doi.org/10.3390/blockchains2040017>
- [9]. Biloshchytskyi, A., Yedilkhan, D., Faizullin, A., Biloshchytska, S., Kuchanskyi, O., & Medetbek, A. (2025). Development of the architecture of a software and hardware complex for data collection and management of smart technologies. In *2025 IEEE 13th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS)* (pp. 1-7). IEEE. <https://doi.org/10.1109/IDAACS68557.2025.11322246>
- [10]. Omirgaliyev, R., Khamzina, A., Yedilkhan, D., Friedrich, D., & Zhakiyev, N. (2025). An advanced data analysis method using heating degree-days for estimating the efficiency of buildings in smart cities: A case study of Astana. *Results in Engineering*, 28, 107502. <https://doi.org/10.1016/j.rineng.2025.107502>
- [11]. Wen, S.-F., & Katt, B. (2024). Ontology-based metrics computation for system security assurance evaluation. *Journal of Applied Security Research*, 19(2), 230–275. <https://doi.org/10.1080/19361610.2022.2157190>
- [12]. El Marzak, Y., Moudoubah, L., Chahid, A., Faris, S., & Mansouri, K. (2026). Designing an ontology-based framework for ISO 27002-based information security risk management. *Engineering, Technology & Applied Science Research*, 16(1), 31741–31747. <https://doi.org/10.48084/etasr.15794>

- [13]. Preuveneers, D., & Joosen, W. (2024). An ontology-based cybersecurity framework for AI-enabled systems and applications. *Future Internet*, 16(3), 69. <https://doi.org/10.3390/fi16030069>
- [14]. Oliveira, Í., Sales, T. P., Almeida, J. P. A., Baratella, R., Fumagalli, M., & Guizzardi, G. (2024). Ontology-based security modeling in ArchiMate. *Software and Systems Modeling*, 23(4), 925–952. <https://doi.org/10.1007/s10270-024-01149-1>
- [15]. Fenz, S., Plieschnegger, S., & Hobel, H. (2016). Mapping information security standard ISO 27002 to an ontological structure. *Information & Computer Security*, 24(5), 452–473. <https://doi.org/10.1108/ICS-07-2015-0030>
- [16]. Wang, Z., Zhu, H., Liu, P., & Sun, L. (2021). Social engineering in cybersecurity: A domain ontology and knowledge graph application examples. *Cybersecurity*, 4, 31. <https://doi.org/10.1186/s42400-021-00094-6>
- [17]. Wilson, R. S. I., Goonetillake, J. S., Indika, W. A., & Ginige, A. (2023). A conceptual model for ontology quality assessment: A systematic review. *Semantic Web*, 14(6), 1051–1097. <https://doi.org/10.3233/SW-233393>
- [18]. Kitsios, F., Chatzidimitriou, E., & Kamariotou, M. (2023). The ISO/IEC 27001 information security management standard: How to extract value from data in the IT sector. *Sustainability*, 15(7), 5828. <https://doi.org/10.3390/su15075828>
- [19]. Chen, Y. (2022). Information security management: Compliance challenges and new directions. *Journal of Information Technology Case and Application Research*, 24(4), 243–249. <https://doi.org/10.1080/15228053.2022.2148979>
- [20]. Kezadri Hamiaz, M., & Driss, M. (2025). Ethereum smart contracts under scrutiny: A survey of security verification tools, techniques, and challenges. *Computers*, 14(6), 226. <https://doi.org/10.3390/computers14060226>

DOI: 10.37943/ZUOX7374**Azamat Baibussinov**

Master's Degree, Department of Technical Support
a.baibussinov@astanait.edu.kz; orcid.org/0000-0002-9714-278X
National Defense University of the Republic of Kazakhstan, Kazakhstan

Kaisarbek Yesbergenov

PhD, Associate Professor, Head of the Doctoral Program
ekb72@mail.ru, orcid.org/0009-0009-3648-5568
National Defense University of the Republic of Kazakhstan, Kazakhstan

Abdykassym Altynbay

Candidate of Technical Sciences
altinbai1955@gmail.com, orcid.org/0009-0008-3009-5190
“Digital Heritage of Eurasia” LLP, Kazakhstan

Gul Jussupova

Candidate of Technical Sciences, Department of Digitalization and Public Services
g.jussupova@sci.gov.kz, orcid.org/0009-0009-7205-3666
Ministry of Science and Higher Education of the Republic of Kazakhstan,
Kazakhstan

Daulet Toibazarov

PhD, Associate Professor, Senior Researcher,
Department of Scientific Information and Innovations
aldar-kosel@yandex.ru, orcid.org/0009-0000-9135-7360
National Defense University of the Republic of Kazakhstan, Kazakhstan

Murat Zhakenov

Candidate of Technical Sciences
muratzhakenov@outlook.com, orcid.org/0009-0005-9672-4365
“Digital Heritage of Eurasia” LLP, Kazakhstan

**RECORD-LEVEL VALIDATION OF OPEN-SOURCE UAV FLIGHT LOG PARSERS
ON THE NIST CFREDS CORPUS**

Abstract: Examiners read drone flight logs with open-source parsers, and published comparisons of these parsers count the rows a tool writes or note whether it crashed. We tested whether such counts say anything about the evidence. Six parsers (DatCon, DROP, DRDP, Fodogu, pymavlink and GRYPHON) were run three times on each of the 72 files of four NIST CFReDS drone datasets, 1,296 runs with pinned versions and hashed inputs and outputs. Every coordinate a tool wrote was compared with two reference decoders: one accepts a DJI packet only when its checksum verifies, the other reproduces the Mission Planner exports shipped with the ArduPilot logs. On the DJI Phantom 3 files, DatCon and DROP wrote no unsupported position and agreed exactly wherever both reported one; 729,793 reference records (82.4%) are reproduced by both. The count differences earlier read as complementary recovery come from repeated rows and from two undocumented display rules: DROP blanks positions logged with two or fewer satellites, and DatCon writes nothing before a flag set by the first record with non-zero position and height. At the rate used here, 98.3% of DatCon's GPS-receiver rows repeat an earlier value (97.9% at its 30 Hz default). Runs that ended with status zero and no coordinates from a file holding them made up 21 of 126 runs on formats a tool claims to support and 61 of 120 on other formats. On the ArduPilot logs, 545 of 927 positions that pass a plain latitude and longitude range test were logged without a three-dimensional fix. All 431 completed run triples were bit-identical. The procedure is stated as a formal model, so that each reported quantity has a definition.