М І Ф
Математика, Інформатика, Фізика: Наука та Освіта, Том 3, № 2 (2026), с. 181–198.
Mathematics, Informatics, Physics: Science and Education, Volume 3, No. 2 (2026), pp. 181–198.
Journal homepage: https://intranet.vspu.edu.ua/miph

UDC 004.056:510.644.4

Monotone risk-adaptive access control for smart university environments with a KSU24 integration design

Oleksandr Lemeshchuk[1], Dmitriy Klionov[2], Denys Senchyshen[3], Rostyslav Zasukhin[4]

[1]Kherson State University,
Department of Computer Science and Software Engineering, Ivano-Frankivsk, Ukraine
Olemeshchuk@ksu.ks.ua
https://orcid.org/0000-0002-9876-3502

[2]Kherson State University,
Department of Informatics, Ivano-Frankivsk, Ukraine
dklonov@ksu.ks.ua
https://orcid.org/0009-0004-8537-3806

[3]Kherson State University,
Department of Computer Science and Software Engineering, Ivano-Frankivsk, Ukraine
dsenchishen@ksu.ks.ua
https://orcid.org/0000-0002-4311-7095

[4]Kherson State University,
Department of Computer Science and Software Engineering, Ivano-Frankivsk, Ukraine
000950@university.kherson.ua
https://orcid.org/0009-0009-8097-3366

Abstract. This paper proposes a risk-adaptive access-control policy and an integration design for KSU24, the digital platform of Kherson State University. A hard authorization gate precedes a fuzzy score based on resource criticality, schedule deviation, time, location or network context, and device distrust. Cumulative memberships and minimum–maximum rules yield a continuous, coordinate-wise non-decreasing score. The construction is supported by a proof, a worked laboratory example, and a proposed device-bound credential protocol.

e-ISSN 3041-1955 DOI: https://doi.org/10.31652/3041-1955-2026-03-02-07
2020 Mathematics Subject Classification: 68M25, 03B52.
© 2026 Lemeshchuk O., Klionov D., Senchyshen D., Zasukhin R., Creative Commons Attribution 4.0 International Licence.

On 100,000 synthetic requests, mean modeled interaction time is 3.206 seconds for the proposed policy, 5.501 seconds for static strong verification, and 3.269 seconds for a crisp policy with matched rules. Both adaptive policies meet the same policy-derived hazardous-set tests. No monotonicity violations occur in 262,440 adjacent grid comparisons. Within a KSU24-only online model, an analytical treatment addresses clock faults, network degradation, token expiry, and nonce reuse. A separate discrete-event experiment uses 30 replications of 10,000 requests in each of ten scenarios. Under severe synthetic network delay, 49.492% of requests obtain a grant; an undetected backward clock step admits 5,204 expired credentials when the monotonic check is removed, and none when it is retained. The experiment is executed in Python; a BPMN timing model is supplied for subsequent Bizagi analysis. These results characterize the specified policy under explicit assumptions; they do not establish attack-detection accuracy, measured hardware performance, or a deployed KSU24 integration.

Keywords: risk-adaptive access control, fuzzy inference, monotonicity, context-aware authorization, KSU24, device-bound credentials.

1. Introduction

University identities connect teaching, administration, laboratories, and shared facilities. The wider integration of physical resources is also studied in smart-city systems [21]. Role-based access control (RBAC) provides stable permissions [1], but identical roles may involve different timetable, device, or network conditions. Attribute-based and context-aware authorization incorporate this evidence [2, 3, 4, 11]. Strong authentication for every interaction can impose repeated effort; conversely, a reusable bearer code may be copied. Authorization, authentication strength, and credential delivery must therefore be treated separately.

Risk-adaptive control selects a response using current evidence [5, 9, 10]. Fuzzy sets [6] support gradual boundaries, and quantified fuzzy risk and security-sensitive inference have been studied previously [7, 8]. However, the triangular membership of the category “medium” decreases beyond its peak. Using it directly as cumulative adverse evidence can lower a score when an anomaly grows. The condition “at least medium” requires a different construction.

Context-aware implementations also address policy distribution and enforcement. Riad and Cheng extend XACML with run-time parameters and access-code adaptation [12]; Kalaria et al. investigate adaptive decisions in fog infrastructure [13]; Kayes et al. study scalability and flexibility in fog-based control [24]. These works establish relevant foundations. The comparison here concerns verification selection under shared synthetic inputs, rather than a reimplementation of their systems or a comparison of backend processing times.

The unresolved application problem is to connect an inspectable, monotone university policy to enforceable transaction-specific verification. The contribution is a particular cumulative fuzzy construction, operational timetable normalization, a proof of monotonicity, and a reproducible synthetic comparison with a crisp translation of the same rules. RBAC, fuzzy inference, monotone aggregation, and strong authentication are established techniques. Neither their novelty nor superiority over all adaptive systems is claimed.

2. Statement of the problem

The objective is to construct a continuous score that cannot decrease when any normalized adverse input increases, map that score to verification requirements, and characterize its behavior under explicit workload assumptions. The enforcement design must bind completed verification to the resource, action, reader challenge, and registered device key.

KSU24 is the digital platform of Kherson State University [23]. Institutional adapters are proposed for identity, role, timetable, and reservation evidence. Public platform information establishes the setting, but does not verify the interfaces assumed here. The policy decision point (PDP), policy enforcement point (PEP), wallet, and approval service are proposed components; no deployment is claimed. The present scope is restricted to KSU24: cross-platform authorization and independent offline validation are outside the model.

An eligible request has normalized context $X = (C, S, T, L, D) \in [0, 1]^5$. Higher values indicate more adverse evidence: resource criticality, schedule deviation, temporal anomaly, location/network anomaly, and device distrust. Table 1 gives candidate operational mappings. Values and thresholds are design choices, not learned probabilities of compromise.

Table 1. Candidate context normalization and evidence sources

Input	Mapping and trusted source
C	Owner-approved resource registry: public 0.10; classroom 0.30; laboratory 0.55; critical control or sensitive data 0.90.
S	$\min(1, \Delta_s/60)$, where Δ_s is minutes outside the nearest expected session or reservation; $S = 1$ if none exists.
T	Institutional clock/calendar: core hours 0; shoulder period 0.35; after hours 0.70; restricted night, weekend, or holiday 0.90.
L	Registered reader and network evidence: trusted 0.05; partial trust 0.40; inconsistent 0.75; untrusted or tampered 0.95.
D	Device registry/posture: registered and recent 0.05; stale 0.40; newly enrolled with unknown posture 0.75. Revocation is a hard denial.

The hard gate checks role permission, account and resource state, revocation, and mandatory timetable constraints. Deny or indeterminate stops processing. If unscheduled access is conditionally permitted, schedule deviation may be scored; otherwise, stronger authentication cannot override the prohibition. Overall enforcement requires

$$\begin{aligned} \text{permit} = \ & \text{hard_gate} \wedge \text{verification_satisfied} \\ & \wedge \text{obligations_satisfied} \wedge \text{credential_valid}\,. \end{aligned} \tag{1}$$

Missing soft context is conservatively mapped or denied under a versioned policy, never silently assigned a benign value. Input provenance, timestamps, and normalization versions are part of the trusted decision evidence.

3. Main mathematical results

Memberships and cumulative rules. Let $\mathrm{clip}(z) = \min(1, \max(0, z))$. The shared low, medium, and high memberships are

$$\begin{aligned} \mu_{\mathrm{Low}}(x) &= 1 - \mathrm{clip}((x - 0.25)/0.25), \\ \mu_{\mathrm{Med}}(x) &= \max(0, 1 - |(x - 0.50)/0.25|), \\ \mu_{\mathrm{High}}(x) &= \mathrm{clip}((x - 0.50)/0.25). \end{aligned} \tag{2}$$

The policy uses $m(x) = 1 - \mu_{\mathrm{Low}}(x)$ and $h(x) = \mu_{\mathrm{High}}(x)$ as cumulative evidence. Both are non-decreasing, and $0 \le h(x) \le m(x) \le 1$. The triangular medium membership describes a linguistic category and is not used as cumulative evidence in the proposed score.

Let $(q_1, q_2, q_3, q_4) = (S, T, L, D)$. The rule activations are

$$\begin{aligned} a_M(X) &= \max_{x \in \{C,S,T,L,D\}} m(x), \\ H_5(X) &= \min(h(S), m(C)), \\ H_6(X) &= \max_{1 \le i < j \le 4} \min(m(q_i), m(q_j)), \\ a_H(X) &= \max\{h(C), h(T), h(L), h(D), H_5(X), H_6(X)\}. \end{aligned} \tag{3}$$

Here H_5 represents severe timetable mismatch on a resource of at least medium criticality. H_6, the second-largest cumulative value among the four contextual inputs, represents two simultaneous anomalies. The score and default verification level are

$$R(X) = 15 + 35a_M(X) + 35a_H(X), \qquad 15 \le R(X) \le 85, \tag{4}$$

$$\ell(X) = \begin{cases} 1, & R(X) < 35, \\ 2, & 35 \le R(X) < 70, \\ 3, & R(X) \ge 70. \end{cases} \tag{5}$$

The constants in (4) are illustrative policy parameters. Increasing a favorable signal cannot cancel a different adverse signal under maximum aggregation, but a faulty high-risk source may consequently cause excessive escalation.

Theorem 1. *The function R defined by (2)–(4) is continuous and coordinate-wise non-decreasing on $[0, 1]^5$. It admits a convex combination of the values 15, 50, and 85. The verification level in (5) is coordinate-wise non-decreasing.*

Proof. Clipping preserves order and continuity. Thus m and h are continuous and non-decreasing. Finite minima and maxima preserve both properties, including the pairwise-minimum representation of H_6. Consequently a_M and a_H have these properties, as does their positively weighted sum in (4). Every high-rule value is at most a_M: this follows from $h(x) \le m(x)$ for the individual inputs, and from the definitions of H_5 and H_6. Hence $0 \le a_H \le a_M \le 1$ and

$$R = 15(1 - a_M) + 50(a_M - a_H) + 85a_H. \tag{6}$$

The three weights in (6) are nonnegative and sum to one. This proves the range and a zero-order Sugeno-style interpretation without requiring another defuzzification operation. Finally, fixed ordered thresholds are non-decreasing functions of R, proving the level assertion. The levels themselves are discontinuous at the thresholds. □

Worked laboratory scenario. Suppose the role and resource policy allow conditional laboratory access outside a scheduled session. Let $C = 0.55$, $T = 0$, and $L = D = 0.05$. Forty minutes after the session, $S = 40/60 = 2/3$. Equation (2) gives $m(C) = m(S) = 1$, $h(C) = 0.2$, $h(S) = 2/3$, and zero cumulative evidence for the other inputs. Thus $a_M = 1$, $a_H = 2/3$, and

$$R = 15 + 35 + 35(2/3) = 73\tfrac{1}{3}, \tag{7}$$

which requires L3. During the session, $S = 0$ yields $a_H = 0.2$ and $R = 57$, requiring L2. Figure 1 shows the continuous score change. Independent approvals remain separate obligations in (1); if unscheduled entry is forbidden, the hard gate denies access before scoring.

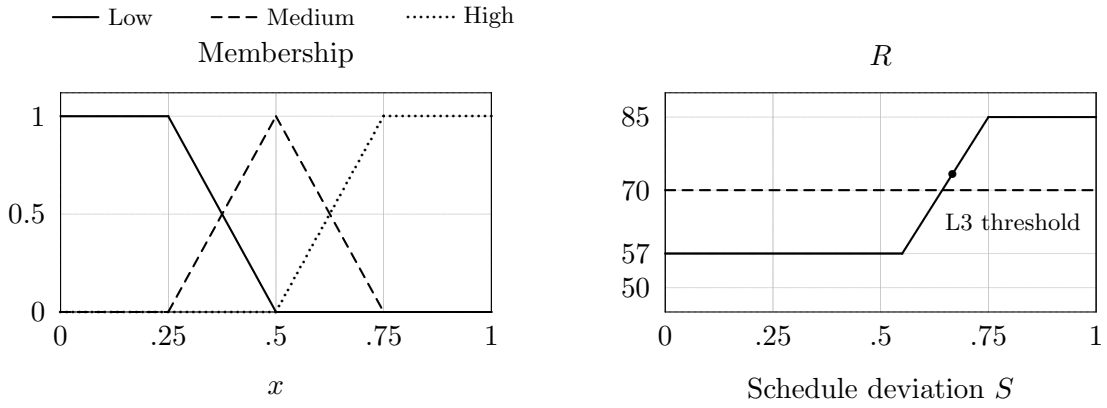

FIGURE 1. Membership partition and the laboratory score as schedule deviation increases

4. Proposed integration and enforcement

4.1. KSU24 scope and authorization exchange. L1 uses an existing authenticated session, a registered application key, and a valid token. L2 requires one user-facing step-up, preferably a user-verifying passkey; an institution-approved one-time-password or push fallback has weaker phishing resistance. L3 requires a validated, user-verifying public-key ceremony with no password-code-only fallback. Local biometric recognition or a PIN may activate an authenticator; biometric templates remain local. These are policy profiles, not claims of NIST authenticator assurance level conformance [15]. A higher resource-specific minimum overrides the score-derived level.

The architecture separates information, decision, and enforcement responsibilities as in XACML [22], and uses explicit verification principles consistent with zero-trust design [14]. Neither an implemented XACML profile nor complete zero-trust conformance is claimed. The PDP, adapters, identity service, device registry, signing keys, readers, and audit service form the trusted computing base. Their channels are mutually authenticated. Signed evidence still depends on the integrity of its source.

The present model is confined to KSU24. A student opens the authenticated personal profile on a phone, where the server-issued credential is displayed as a QR code. The profile acts as the client interface; the phone does not determine the authoritative issue or expiry time. A reader forwards the presentation to the KSU24 backend through an authenticated channel. Issuance and final validation are assumed to share one authoritative server-side time

domain and transaction state. PDP and PEP denote separate logical responsibilities within that domain. Being part of KSU24 alone does not guarantee synchronized clocks; the shared validation domain is an explicit model assumption. Phone and reader clocks are not used to decide token validity, and readers do not issue independent offline grants.

The online exchange proceeds as follows. The backend PEP generates a random 128-bit nonce and binds it to the transaction, reader, resource, and action. The PDP checks the hard gate, collects time-bounded context, and selects verification and approval obligations. After their completion it rechecks authorization, revocation, context freshness, and the policy version. A material change cancels the pending decision. The issuer then returns a signed credential to the wallet, with a proposed lifetime of 30 seconds from issuance. The profile presents it for the intended reader transaction together with proof of the registered application key, and the backend PEP performs the final validation. The PEP validates the signatures and transaction bindings and atomically consumes the token identifier before granting access.

The token uses COSE_Sign1 [17] with ES256 [26], a protected algorithm identifier, and an institutional trusted-key registry. Its payload is a deterministic CBOR map [25] containing issuer and audience; pseudonymous subject and application-key reference; resource and action; completed level and decision identifier; issue and expiry times; server generation identifier; unique token identifier; reader nonce; context digest; and policy version. These are application claims, not a claim of a fully registered token profile. Device proof signs the deterministic CBOR array

$$[\texttt{"KSU24-POP-v1"},\ \mathrm{SHA256}(\mathrm{token}),\ \mathrm{reader_nonce}].$$

The reader challenge is delivered to the profile over an authenticated channel before issuance. A QR image alone does not prove possession of the registered application key; the signed device proof must accompany the same transaction. QR and NFC are presentation transports, not authentication factors. A synchronized passkey does not identify one physical device, so device proof uses the separate application key.

Multi-party authorization is independent of the verification level and follows separation of duty [19]. Policy specifies a threshold, distinct eligible approvers, and any role quotas. Requester self-approval, duplicates, expired approvals, and mismatched decisions are rejected. A challenge hashes the decision, requester, resource, action, context digest, policy version, approval nonce, and expiry, using deterministic CBOR and a distinct `KSU24-APPROVE-v1` tag. Each approver completes a full WebAuthn ceremony; the service checks the challenge, origin, relying-party identifier, signature, and user verification [16]. The proposed approval window is 120 seconds; the separate 30-second token lifetime starts only after final rechecks and issuance.

4.2. Clock faults and token expiry. Shared server-side time removes the need to compare the phone or reader clock with the issuing clock. It does not remove changes in the server clock or the need for correct calendar time in timetable-based decisions. Let t_i and t_v denote real issuance and validation times, and write the server clock as $c(t) = t + e(t)$. For the proposed lifetime $\tau = 30$ seconds, checking $c(t_v) < c(t_i) + \tau$ is equivalent to

$$t_v - t_i < \tau + e(t_i) - e(t_v).$$

A constant common offset cancels when determining token age. A backward clock step may extend the apparent validity of an unused token; a forward step may reject it early. Even a constant offset can invalidate timetable decisions if absolute institutional time is wrong.

The model therefore requires an additional elapsed-time deadline for each pending transaction and issued token, measured by a monotonic clock within one uninterrupted server generation. Acceptance requires both the timestamp check and this deadline to pass. The argument assumes continuity and bounded rate error of the elapsed-time clock; monotonicity alone does not establish accurate calendar time. A detected time discontinuity or loss of confidence in required time evidence suspends affected new grants. Missing time evidence is not mapped to a benign risk value. Loss of external synchronization alone need not imply an immediate clock error, but continued acceptance requires the relevant time-error bounds to remain trustworthy.

Restart or loss of timing continuity invalidates outstanding credentials and challenges from the previous generation. The active generation and replay state must remain authoritative across recovery; restoring an old snapshot cannot make a consumed token acceptable again. If that state is uncertain, new grants remain blocked. These are analytical conditions of the model, not demonstrated availability guarantees.

No additional expiry leeway is assumed in this centralized profile. A fixed ± 2-second tolerance is not needed to reconcile phone or reader clocks here. Independent validation clocks would require a separate model with bounded clock disagreement and a replay-retention window covering every permitted acceptance time; that case lies outside the present KSU24-only profile.

4.3. Network degradation and delayed presentations. Network degradation includes increased or variable delay, packet loss, interrupted connections, and reduced service capacity. Denial of service may cause these effects deliberately, while ordinary congestion can produce similar symptoms. Keeping issuance and validation inside KSU24 does not eliminate communication time between the profile, reader, and backend. Queueing and repeated transmission can make an otherwise legitimate token expire.

Expiry is checked at final server-side validation, after queueing and any wait for access to shared transaction state, immediately before consumption. For illustration, a token issued at time 0 and expiring at 30 is scanned at elapsed time 29 but reaches final validation at 31. Under stable clocks it is rejected. These values illustrate the rule and are not measurements. A client-reported scan timestamp cannot extend the token lifetime or move validation into the past; even a signed timestamp does not establish the accuracy of the originating clock.

After expiry, a fresh challenge and authorization decision are required. Network delay does not automatically extend the 30-second lifetime, reduce L2 or L3 requirements, or bypass current revocation checks. This protects the stated acceptance conditions at the cost of refusing some legitimate requests during disruption. Temporary service unavailability is recorded separately from denial caused by insufficient permissions.

4.4. Nonce state, duplicates, and lost responses. A nonce is an unpredictable challenge bound to a transaction, reader, resource, and action; it is not a timestamp. Its validity is determined by authoritative server-side state and a bounded deadline. The state distinguishes pending, consumed, expired, and cancelled transactions. A replacement challenge cancels the earlier pending one. Both token and operation identifiers must be protected against concurrent reuse, so credential reissuance cannot create two grants for the same operation. Related distinctions between server-provided challenges, time windows, and replay tracking appear in RFC 9449 [27]; the COSE protocol considered here is not an OAuth DPoP profile.

The final checks and the transition from pending to consumed must form one guarded authorization operation: concurrent duplicate presentations can produce at most one committed grant. If the response is lost after that commitment, a retry may retrieve the existing result through an authenticated status request but cannot create a second grant. If the outcome cannot be established, it remains indeterminate. Unavailable replay state prevents new authorization; an isolated cache cannot independently accept a token. Replay records are retained for the full acceptance window, and consumed operation state is preserved for the permitted recovery period.

4.5. Mass failures and recovery. Simultaneous client clock errors do not change token acceptance under the shared-server assumption. A fault in the common server time, authorization service, or replay state can affect all dependent requests. The model therefore separates preservation of authorization conditions from service availability: refusing a request under uncertainty prevents an unchecked grant but does not preserve availability.

Bounded queues, limits on pending requests, finite timeouts, and a bounded number of retries with increasing pauses and random jitter limit additional load caused by repeated requests. Retries remain within the original transaction deadline. These controls do not guarantee resistance to every denial-of-service attack. A large backlog cannot justify accepting expired credentials or lowering verification requirements. Recovery requires trustworthy time and consistent authorization and replay state; expired requests are discarded, while uncertain completed operations are reconciled before new grants are considered.

Table 2 summarizes the analytical cases. It states required behavior under the model assumptions, not observed outcomes.

Table 2. Analytical behavior under clock and network failures

Failure	Acceptance rule	Recovery condition
Phone or reader clock offset	Server-side time remains authoritative	No acceptance based on client time
Server clock discontinuity	Suspend affected grants; reject old generations	Restore trustworthy time and state
Delay beyond token expiry	Reject the late presentation	Obtain a fresh challenge and decision
Concurrent duplicate requests	At most one committed grant	Preserve the recorded outcome
Response lost after commitment	Do not authorize a second operation	Retrieve and reconcile existing status
Unavailable authorization or replay state	No new grant without required evidence	Restore authoritative state
Mass overload or network partition	Bound requests and retries; preserve all checks	Discard expired backlog and reconcile state

Changing the presentation channel does not resolve a shared backend outage. NFC may replace optical QR capture only while preserving the same authorization and replay checks; it does not remove the required network or time dependency. An offline acceptance path is outside the present model.

4.6. Residual threats and data protection. The adversary may copy a QR image, steal a password, present an unknown device, manipulate untrusted context, or reuse

a credential. Transaction binding, device proof, freshness checks, and atomic consumption address defined replay conditions. Research on QR phishing concerns malicious redirection [18]; it does not validate this protocol. Live relay remains possible without a validated proximity mechanism. Reader compromise, central signing-key compromise, authorized collusion, denial of service, and physical safety require additional controls. At-most-once authorization does not guarantee successful actuation. Unavailable required policy, revocation, or replay services deny new grants; emergency egress is a separate engineering requirement.

Only decision-relevant context should be collected. Audit records include normalized inputs, freshness, policy versions, score, verification outcome, and reason codes, excluding authentication secrets. Pseudonyms and context digests do not automatically anonymize records. Coarse location, justified retention, separated re-identification mappings, and accessible biometric-free verification are necessary design considerations. GDPR principles provide a reference [20]; applicability and lawful basis depend on the deployment.

5. Synthetic evaluation

Workload and reproducibility. The experiment uses $N = 100\,000$ independently generated, RBAC-eligible requests. It contains no operational KSU24 logs, personal data, or measured authenticator times. Table 3 defines the three-component mixture for each input; $\mathrm{Tri}(a, b, c)$ denotes a triangular distribution with minimum a, mode b, and maximum c. The samples are continuous synthetic values, not measurements of the mappings in Table 1.

Table 3. Synthetic marginal distributions

Input	Probabilities	Component triples (a, b, c) in sampling order
C	0.65; 0.25; 0.10	$(.05, .18, .40)$; $(.40, .55, .72)$; $(.78, .90, 1)$
S	0.80; 0.14; 0.06	$(0, .05, .24)$; $(.26, .42, .60)$; $(.62, .82, 1)$
T	0.88; 0.09; 0.03	$(0, .08, .28)$; $(.30, .45, .62)$; $(.65, .85, 1)$
L	0.95; 0.04; 0.01	$(0, .05, .25)$; $(.28, .45, .62)$; $(.68, .85, 1)$
D	0.93; 0.06; 0.01	$(0, .05, .25)$; $(.30, .48, .65)$; $(.70, .88, 1)$

A NumPy Generator with PCG64 seed 24062026 samples the five columns in order: C, S, T, L, and D. For each column it draws all N labels with the stated probabilities, then triangular values for classes 0, 1, and 2, assigning each class to its selected rows in ascending row order. Computation uses float64. The little-endian C-order $N \times 5$ matrix has SHA-256 digest

```
5c37fceccc62f9920f4cb0af1461f49ea69
6c6e3951ffe137a628850eb9bd5e0
```

The two displayed lines form one uninterrupted digest. It identifies the numerical workload, not a relationship to university traffic. The reference calculation was reconstructed from the specified model during manuscript verification; it is not an existing KSU24 implementation.

The same generator then draws complete timing arrays $T_s \sim \mathrm{Tri}(1, 1.5, 2)$, $T_2 \sim \mathrm{Tri}(2, 2.5, 3)$, and $T_3 \sim \mathrm{Tri}(3, 4, 5)$ seconds, in that order. Every policy reuses the same arrays. L2 and L3 are alternative ceremonies, yielding

$$T_{\mathrm{auth}} = T_s + \mathbf{1}[\ell = 2]T_2 + \mathbf{1}[\ell = 3]T_3. \tag{8}$$

The population mean under these assumptions is $1.5 + 2.5p_2 + 4p_3$ seconds. The model excludes backend computation, context retrieval, network delay, approval waiting, failed

ceremonies, and physical actuation. Confidence intervals use $\bar{t} \pm 1.96s/\sqrt{N}$, with sample standard deviation; differences use paired request-level observations. These are Monte Carlo intervals conditional on the assumed model.

Comparators and outcomes. Static L1 and static L3 assign one level to every eligible request. The matched crisp policy replaces $m(x)$ by $\mathbf{1}[x \geq 0.375]$ and $h(x)$ by $\mathbf{1}[x \geq 0.625]$ in *every* rule of (3). A high rule selects L3; otherwise a medium rule selects L2, with L1 as the default. The fuzzy policy uses (5), including equality at both lower boundaries. These comparators share permissions, rule semantics, inputs, and timing variates; static L3 does not represent all multifactor-authentication deployments.

Table 4. Verification shares and primary modeled interaction times

Policy	L1, %	L2, %	L3, %	Mean, s	P95, s
Static L1	100.00	0.00	0.00	1.500	1.844
Static L3	0.00	0.00	100.00	5.501	6.259
Crisp	42.67	35.01	22.32	3.269	5.869
Proposed	44.01	35.67	20.32	3.206	5.835

Table 4 shows that the proposed policy assigns 44,011, 35,673, and 20,316 requests to L1, L2, and L3. The 95% intervals for mean time are $[3.195, 3.216]$ seconds for the proposed policy, $[3.259, 3.280]$ for crisp, $[5.499, 5.504]$ for static L3, and $[1.499, 1.501]$ for static L1. Relative mean reductions are 41.7% against static L3 and 1.95% against crisp. The paired reduction against crisp is 0.0637 seconds, with interval $[0.0615, 0.0660]$. This is a small conditional difference, not demonstrated practical acceleration. Additional ceremonies average 0.560 per proposed request and 0.573 per crisp request.

An alternative profile adds T_2 whenever $\ell \geq 2$ and T_3 when $\ell = 3$, requiring two sequential ceremonies for L3. The corresponding means are 3.714, 3.828, and 8.003 seconds for proposed, crisp, and static L3. The larger relative reductions, 3.0% and 53.6%, concern a different interaction protocol. A ceremony may use multiple factors, so ceremony count is not factor count.

Policy consistency and monotonicity. The hazardous test set H comprises requests satisfying at least one of

$$C \geq .75, \quad T \geq .75, \quad L \geq .75, \quad D \geq .75,$$
$$(S \geq .75 \ \wedge \ C \geq .50), \qquad \#\{i \in \{1,2,3,4\} : q_i \geq .50\} \geq 2.$$

The benign plateau B has every input at most 0.25. Here $|H| = 15\,275$ and $|B| = 28\,051$. Every predicate defining H forces $a_M = a_H = 1$ and $R = 85$, and also activates a crisp high rule. Both adaptive policies therefore assign all H requests to L3 and none of B above L1. This checks consistency with the policy definition; it cannot estimate attack-detection accuracy or false-negative rates.

A nine-point Cartesian grid, $\{0, .125, \ldots, 1\}^5$, contains 59,049 points and 262,440 adjacent comparisons. Both the proposed score and crisp levels have zero decreases. This finite check supports implementation verification; Theorem 1 supplies the continuous guarantee.

For a controlled ablation, replace every $m(x)$ in (3) with $\mu_{\mathrm{Med}}(x)$, retaining h and (4). This substitution produces 28,166 decreasing transitions (10.73%) and a maximum drop of 35. With $S = .75$ and $T = L = D = 0$, increasing C from .50 to .625 reduces the ablated score

from 85 to 50. The test isolates an unsafe substitution; it is not a representative Mamdani system or a reproduction of prior work.

Noise, calibration, and dependence. Ten independent Gaussian perturbation matrices with standard deviation .03 use a separate PCG64 seed 24062027. Perturbed inputs are clipped to $[0, 1]$, and each matrix is shared by both adaptive policies. Changed levels average 4.0718% for the proposed policy and 4.5225% for crisp, with sample standard deviations 0.0688 and 0.0590 percentage points. Constant policies never change, showing why stability alone is not a quality criterion.

Changing the level thresholds from 30/65 to 40/75 lowers the proposed L3 share from 23.84% to 17.92% and mean modeled time from 3.333 to 3.096 seconds. Shifting all three membership breakpoints by -0.05 or $+0.05$ gives means of 3.397 or 3.038 seconds. These are separate one-factor tests. With the fixed original H, coverage remains 100%: baseline memberships give $R = 85$; even the positive breakpoint shift gives a minimum of at least 71 on H, above the unchanged threshold 70.

In a dependence stress test, the sampled T values are reordered to match the ascending ranks of S, preserving both marginals. The proposed L3 share rises to 23.17%, while L1 also rises to 49.26% because adverse evidence becomes concentrated in fewer requests. Mean time is 3.116 seconds for proposed and 3.185 seconds for crisp; both still cover H, now containing 16,547 requests. This artificial dependence is not a measured campus scenario. It shows that the decision distribution depends materially on workload assumptions.

6. Discrete-event simulation of KSU24 failure scenarios

Scope and reproducible execution. To extend the failure analysis in Section 4, a finite-horizon discrete-event model was executed independently in Python. It is not a KSU24 implementation, a hardware experiment, or a run of Bizagi Modeler. The accompanying BPMN 2.0 file represents the timing and resource path for subsequent use in Bizagi. Bizagi supports BPMN import, but its simulation tokens do not carry associated data [28, 29]. Accordingly, individual token ages, generations and nonce states are evaluated in the Python model rather than approximated by random expiry or replay gateways. Import and execution in Bizagi have not been verified in this study.

The source workload of Section 5 was reconstructed with the same SHA-256 digest and level counts. Each new request independently draws L1/L2/L3 with probabilities 0.44011/ 0.35673/0.20316 and uses the same ceremony distributions as (8). This resamples the level distribution, not the original context records. All requests are RBAC-eligible and satisfy their selected ceremony, signature and device-proof requirements. Cryptography and risk-context acquisition are not executed. Approval queues, repeated-user dependence and physical actuation are excluded.

Each scenario contains 30 independent replications of 10,000 logical requests, starting empty and processing all scheduled events after the last arrival. No warm-up observations are removed: the experiment describes a finite workload, not steady-state operation. NumPy PCG64 uses seed $20261031 + r$ for replication $r = 0, \ldots, 29$; fixed-order random arrays are shared across scenarios. The package contains the runnable source, configuration, all 300 replication records and a sample event trace. Thus the ten scenarios represent three million simulated request instances, with deliberately paired inputs across scenarios.

Timing, state and fault assumptions. Unless varied, arrivals form a Poisson process at 10 requests/s; four identical validation workers serve a FIFO queue of at most 1,000 waiting submissions. Token issuance after the ceremony takes an assumed 0.02 s. Transmission to KSU24 takes Tri(.05, .10, .20) s, and a worker takes Tri(.02, .04, .08) s. These new durations and capacities are scenario assumptions, not measured KSU24 characteristics. The 30 s lifetime starts after issuance. Final expiry is checked after queueing and service; even a request already expired at service start incurs the full assumed service time. This conservative queue discipline does not implement early eviction of stale work.

Each logical operation has an issuance time, generation and consumed flag. The final check and the transition to consumed are one atomic simulated event. A repeated presentation of a committed operation returns its previous status without a new grant. Wall time follows $c(t) = t + e(t)$; the protected model also requires monotonic elapsed time to be strictly less than 30 s. Monotonic time is ideal in this model: drift and monotonic-clock failure are not simulated. Health pauses block issuance and validation, and recovery changes the generation. The health detector, durable-state restoration and atomicity are assumptions, not tested components.

Table 5. Discrete-event scenarios; unspecified parameters equal A

ID	Changed conditions
A	Nominal network and four validation workers.
B	Delivery Tri(.5, 2, 8) s; independent 10% submission loss; at most one retry after $10 + U(.1, .5)$ s if no reply, using the same token.
C	Severe delivery delay Tri(20, 30, 40) s; no loss or retry.
D	Arrival rate 30/s; one validation worker; unchanged queue limit.
E	Backend unavailable during [300, 360) s; generation changes at recovery.
F	Delay as C; an undetected −20 s server wall-clock step at $t = 300$ s; both expiry checks retained.
G	Same as F, but the monotonic expiry check is removed (unsafe ablation).
H	The −20 s step is detected immediately; grants blocked during [300, 305) s; new generation at 305 s; nominal delivery.
I	Loss/retry as B with nominal delivery; 10% of operations receive one extra delivered token copy; 20% of grant replies are lost. A reliable status query follows after 1 s plus transmission.
J	Endpoint clock offsets ±300 s; endpoint time is not a validation input.

Two lost submissions are classified as unavailable at the original deadline, not as an authorization denial. Outage-time arrivals are rejected rather than buffered for recovery. The extra copies in I are deliberately delivered independently of the ordinary submission-loss draws. Status queries consume the same worker resource and never create grants; their delivery is assumed reliable in I. No retry extends token lifetime or lowers the selected verification level.

Results and interpretation. Table 6 reports means across replications. The 95% Monte Carlo interval is $\bar{x} \pm t_{.975,29} s_x / \sqrt{30}$; it quantifies sampling variability under the specified model, not uncertainty in real network conditions. Grant latency is measured from request arrival before the ceremony to the server's granting decision, conditional on a grant. It excludes receipt of the final response and can exceed 30 s because the ceremony precedes token issuance. Degenerate intervals reflect zero observed between-replication variance, not a population-level security guarantee.

TABLE 6. Python discrete-event results, 300,000 requests per scenario

ID	Grants, % (95% interval)	Mean latency, s	Expired grants	Extra grants
A	100.000 (100.000–100.000)	3.389	0	0
B	99.012 (98.972–99.053)	7.709	0	0
C	49.492 (49.312–49.672)	29.912	0	0
D	22.650 (22.169–23.130)	17.940	0	0
E	94.035 (93.959–94.111)	3.390	0	0
F	49.492 (49.312–49.672)	29.912	0	0
G	51.226 (51.036–51.416)	30.144	5 204	0
H	99.474 (99.448–99.501)	3.389	0	0
I	99.107 (99.068–99.147)	4.230	0	0
J	100.000 (100.000–100.000)	3.389	0	0

Nominal mean grant latency is 3.389 s (95% interval 3.383–3.394), compared with the 3.206 s ceremony-only sample mean in Section 5. These are different workloads and timing scopes, not a measured implementation overhead. In B, one bounded retry yields a 99.012% grant fraction, while double losses leave 2,963 operations unavailable. Mean successful latency increases to 7.709 s. In C, 151,525 requests expire and 148,475 obtain grants. The lower latency reported only among successful requests must not be interpreted as the experience of all users.

Under overload D, 67,949 requests obtain grants, 176,630 expire and 55,421 are rejected by the queue limit. Mean worker utilization over the event-observation interval is 99.287%. Hence expiry preservation can coexist with severe availability loss. In E, 17,895 requests are unavailable during the modeled interruption. H produces 1,577 unavailable requests; its short network delay creates no old-generation rejection in the random sample, so generation invalidation is checked separately by a deterministic test. Neither scenario validates a recovery implementation.

F produces exactly the same outcomes as C because the monotonic bound prevents the backward step from extending lifetime. Removing that bound in G produces 5,204 grants with actual token age at least 30 s: 1.735% of all requests. The apparent increase in grants is a safety violation, not improved availability. This ablation specifically assumes an undetected wall-clock fault and isolates the additional monotonic check.

In I, all 59,372 lost grant replies are resolved by the assumed reliable status-query path; 26,992 repeated token presentations return an existing result. There are no additional grants for an already consumed operation in any scenario. J matches A exactly because endpoint clocks are excluded by construction. Deterministic tests cover equality at expiry, delayed final checks, backward and forward wall-clock steps, unknown ledger state, generation changes and two coincident presentations. These results verify consistency of the stated simulation rules. They do not establish concurrent database correctness, resistance to real attacks or a validated clock-health detector.

BPMN model and Bizagi reports. The supplied BPMN timing path contains common verification, alternative L2/L3 ceremonies, token issuance, transmission, and final validation service. Exclusive routing uses the level shares from Section 5. In Bizagi, the guide specifies nominal, delayed-network and reduced-capacity timing scenarios, 30 replications, an explicit duration, and the validation-worker resource. Replications are run through What-If Analysis; results can be exported to Excel [29, 30]. The end event denotes a processed

attempt, not a permission grant. The diagram does not implement the Python model's loss, bounded queue, token deadlines or replay state, and their outcome counts must not be equated. Resource utilization also requires matching observation durations because Bizagi includes the configured idle tail [30]. No numerical result in this paper is attributed to a Bizagi execution.

7. Discussion and conclusions

The principal result is an inspectable continuous score with a mathematical monotonicity guarantee, coupled to a transaction-bound enforcement design. The matched crisp policy is also monotone and meets the same policy-derived hazardous and benign tests. The small modeled time difference therefore arises from decision boundaries and assumed ceremony costs, rather than an exclusive safety property.

The evaluation omits repeated-user histories, arrival bursts, realistic context dependence, accessibility effects, and independently labeled adversarial requests. Section 6 models selected outages and recovery rules but does not validate real fault detection, durable recovery, or cryptographic enforcement. Resource values, memberships, and delays are illustrative, without observed calibration or expert-elicitation data. Claims about incident reduction, false acceptance, fairness, deployed latency, or physical security would require additional evidence.

The analysis of clock and network failures is restricted to the KSU24-only online model. Shared server-side validation reduces dependence on endpoint clocks, while server time discontinuities, delayed requests, and uncertain replay state remain explicit failure cases. Token expiry, one-time consumption, and verification requirements are preserved under the stated assumptions; availability may be lost when the necessary evidence is unavailable. The event simulation quantifies conditional availability losses and illustrates an unsafe clock-check ablation. It preserves the mathematical results and the original synthetic evaluation in Section 5; neither the Python results nor the BPMN artifact demonstrate a deployed KSU24 system.

Conflict of interests and ethics. The authors disclose their affiliation with Kherson State University, whose KSU24 platform is the institutional setting of the proposed integration. The authors declare no other conflicts of interest. The study uses synthetic contexts, assumed interaction times, and simulated request events. No human participants, personal records, biometric samples, operational logs, or hardware measurements were used.

Acknowledgements. The authors declare no special funding for this work.

References

1. Sandhu, R. S., Coyne, E. J., Feinstein, H. L., Youman, C. E. (1996). *Role-based access control models*, Computer, **29** (2), 38–47. https://doi.org/10.1109/2.485845
2. Hu, V. C., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., Miller, R., Scarfone, K. (2014). *Guide to attribute based access control (ABAC) definition and considerations* (NIST SP 800-162), National Institute of Standards and Technology, Updated August 2, 2019. https://doi.org/10.6028/NIST.SP.800-162
3. Kulkarni, D., Tripathi, A. R. (2008). *Context-aware role-based access control in pervasive computing systems*, Proceedings of the 13th ACM Symposium on Access Control Models and Technologies, 113–122. https://doi.org/10.1145/1377836.1377854
4. Kayes, A. S. M., Kalaria, R., Sarker, I. H., Islam, M. S., Watters, P. A., Ng, A., Hammoudeh, M., Badsha, S., Kumara, I. (2020). *A survey of context-aware access control mechanisms for cloud and fog networks: Taxonomy and open research issues*, Sensors, **20** (9), Article 2464. https://doi.org/10.3390/s20092464

5. Atlam, H. F., Azad, M. A., Alassafi, M. O., Alshdadi, A. A., Alenezi, A. (2020). *Risk-based access control model: A systematic literature review*, Future Internet, **12** (6), Article 103. https://doi.org/10.3390/fi12060103
6. Zadeh, L. A. (1965). *Fuzzy sets*, Information and Control, **8** (3), 338–353. https://doi.org/10.1016/S0019-9958(65)90241-X
7. Cheng, P.-C., Rohatgi, P., Keser, C., Karger, P. A., Wagner, G. M., Reninger, A. S. (2007). *Fuzzy multi-level security: An experiment on quantified risk-adaptive access control*, Proceedings of the IEEE Symposium on Security and Privacy, 222–230. https://doi.org/10.1109/SP.2007.21
8. Ni, Q., Bertino, E., Lobo, J. (2010). *Risk-based access control systems built on fuzzy inferences*, Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, 250–260. https://doi.org/10.1145/1755688.1755719
9. Kandala, S., Sandhu, R., Bhamidipati, V. (2011). *An attribute based framework for risk-adaptive access control models*, Proceedings of the 6th International Conference on Availability, Reliability and Security, 236–241. https://doi.org/10.1109/ARES.2011.41
10. Shaikh, R. A., Adi, K., Logrippo, L. (2012). *Dynamic risk-based decision methods for access control systems*, Computers & Security, **31** (4), 447–464. https://doi.org/10.1016/j.cose.2012.02.006
11. Servos, D., Osborn, S. L. (2017). *Current research and open problems in attribute-based access control*, ACM Computing Surveys, **49** (4), Article 65. https://doi.org/10.1145/3007204
12. Riad, K., Cheng, J. (2021). *Adaptive XACML access policies for heterogeneous distributed IoT environments*, Information Sciences, **548**, 135–152. https://doi.org/10.1016/j.ins.2020.09.051
13. Kalaria, R., Kayes, A. S. M., Rahayu, W., Pardede, E., Shahraki, A. S. (2024). *Adaptive context-aware access control for IoT environments leveraging fog computing*, International Journal of Information Security, **23**, 3089–3107. https://doi.org/10.1007/s10207-024-00866-4
14. Rose, S., Borchert, O., Mitchell, S., Connelly, S. (2020). *Zero trust architecture* (NIST SP 800-207), National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-207
15. Temoshok, D., Fenton, J. L., Choong, Y.-Y., Lefkovitz, N., Regenscheid, A., Galluzzo, R., Richer, J. P. (2025). *Digital identity guidelines: Authentication and authenticator management* (NIST SP 800-63B-4), National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-63B-4
16. World Wide Web Consortium (2026, August 25). *Web authentication: An API for accessing public key credentials—Level 3* (W3C Recommendation). https://www.w3.org/TR/2026/REC-webauthn-3-20260825/
17. Schaad, J. (2022). *CBOR object signing and encryption (COSE): Structures and process* (RFC 9052), RFC Editor. https://www.rfc-editor.org/rfc/rfc9052.html
18. Geisler, M., Pöhn, D., Hommel, W. (2024). *Hooked: A real-world study on QR code phishing*, Sicherheit in vernetzten Systemen: 31. DFN-Konferenz, B1–B18. epubli. https://dpoehn.de/en/publications/
19. Simon, R. T., Zurko, M. E. (1997). *Separation of duty in role-based environments*, Proceedings of the 10th Computer Security Foundations Workshop, 183–194. https://doi.org/10.1109/CSFW.1997.596811
20. European Parliament and Council of the European Union (2016). *Regulation (EU) 2016/679 (General Data Protection Regulation)*, Official Journal of the European Union, L **119**, 1–88. https://eur-lex.europa.eu/eli/reg/2016/679/oj
21. Zanella, A., Bui, N., Castellani, A. P., Vangelista, L., Zorzi, M. (2014). *Internet of Things for smart cities*, IEEE Internet of Things Journal, **1** (1), 22–32. https://doi.org/10.1109/JIOT.2014.2306328
22. OASIS (2013, January 22). *eXtensible Access Control Markup Language (XACML) version 3.0* (OASIS Standard). https://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html
23. Kherson State University (n.d.). *KSU24*, Retrieved September 29, 2026. https://ksu24.kspu.edu/
24. Kayes, A. S. M., Rahayu, W., Watters, P. A., Alazab, M., Dillon, T., Chang, E. (2020). *Achieving security scalability and flexibility using fog-based context-aware access control*, Future Generation Computer Systems, **107**, 307–323. https://doi.org/10.1016/j.future.2020.02.001
25. Bormann, C., Hoffman, P. (2020). *Concise binary object representation (CBOR)* (RFC 8949), RFC Editor. https://www.rfc-editor.org/rfc/rfc8949.html
26. Schaad, J. (2022). *CBOR object signing and encryption (COSE): Initial algorithms* (RFC 9053), RFC Editor. https://www.rfc-editor.org/rfc/rfc9053.html

27. Fett, D., Campbell, B., Bradley, J., Lodderstedt, T., Jones, M., Waite, D. (2023). *OAuth 2.0 demonstrating proof of possession (DPoP)* (RFC 9449), RFC Editor. https://www.rfc-editor.org/rfc/rfc9449.html
28. Bizagi (n.d.). *Import a diagram from BPMN*, Bizagi Modeler documentation, Retrieved October 31, 2026. https://help.bizagi.com/platform/en/import_diagram_from_bpmn.htm
29. Bizagi (n.d.). *Simulation in Bizagi*, Bizagi Modeler documentation, Retrieved October 31, 2026. https://help.bizagi.com/platform/en/simulation_in_bizagi.htm
30. Bizagi (n.d.). *Configure Simulation*, Bizagi Modeler documentation, Retrieved October 31, 2026. https://help.bizagi.com/platform/en/configure_simulation.htm

УДК 004.056:510.644.4

Монотонне ризик-адаптивне керування доступом у розумному університетському середовищі з проєктом інтеграції KSU24

Олександр Лемещук, Дмитро Кльонов, Денис Сенчишен, Ростислав Засухін

Анотація. Запропоновано ризик-адаптивну політику керування доступом і проєкт її інтеграції з KSU24, цифровою платформою Херсонського державного університету. Перед нечітким оцінюванням виконуються обов'язкові перевірки повноважень. Оцінка враховує критичність ресурсу, відхилення від розкладу, час, контекст місця або мережі та недовіру до пристрою. Кумулятивні функції належності й операції мінімуму та максимуму забезпечують неперервність і неспадання оцінки за кожною координатою. Наведено математичне доведення, приклад доступу до лабораторії та проєкт протоколу облікових даних із прив'язкою до пристрою. На 100 000 синтетичних запитів середній модельований час взаємодії становить 3,206 с для запропонованої політики, 5,501 с для статичної посиленої перевірки та 3,269 с для дискретної політики з тими самими правилами. Обидві адаптивні політики задовольняють однакові тести, побудовані з правил політики. У 262 440 сусідніх порівняннях на сітці порушень монотонності не виявлено. У межах онлайн-моделі, обмеженої KSU24, аналітично розглянуто збої годинників, деградацію мережі, завершення строку дії токенів і повторне використання nonce. Окремий дискретно-подієвий експеримент містить 30 повторень по 10 000 запитів у кожному з десяти сценаріїв. За значної синтетичної затримки мережі доступ отримують 49,492% запитів. Невиявлений зсув годинника назад дає 5 204 видачі за простроченими токенами без монотонної перевірки та жодної з нею. Розрахунки виконано в Python; BPMN-модель часу надано для подальшого аналізу в Bizagi. Результати характеризують визначену політику за явних припущень і не підтверджують точності виявлення атак, апаратної продуктивності чи впровадження в KSU24.

Ключові слова: ризик-адаптивне керування доступом, нечітке виведення, монотонність, контекстно-залежна авторизація, KSU24, облікові дані з прив'язкою до пристрою.

Список використаних джерел

1. Sandhu R. S., Coyne E. J., Feinstein H. L., Youman C. E. Role-based access control models. *Computer*. 1996. Vol. 29, No. 2. P. 38–47. DOI: https://doi.org/10.1109/2.485845.
2. Hu V. C., Ferraiolo D., Kuhn R., Schnitzer A., Sandlin K., Miller R., Scarfone K. Guide to attribute based access control (ABAC) definition and considerations. NIST SP 800-162. National Institute of Standards and Technology, 2014. Updated August 2, 2019. DOI: https://doi.org/10.6028/NIST.SP.800-162.
3. Kulkarni D., Tripathi A. R. Context-aware role-based access control in pervasive computing systems. *Proceedings of the 13th ACM Symposium on Access Control Models and Technologies*. ACM, 2008. P. 113–122. DOI: https://doi.org/10.1145/1377836.1377854.

4. Kayes A. S. M., Kalaria R., Sarker I. H., Islam M. S., Watters P. A., Ng A., Hammoudeh M., Badsha S., Kumara I. A survey of context-aware access control mechanisms for cloud and fog networks: Taxonomy and open research issues. *Sensors*. 2020. Vol. 20, No. 9. Art. 2464. DOI: https://doi.org/10.3390/s20092464.
5. Atlam H. F., Azad M. A., Alassafi M. O., Alshdadi A. A., Alenezi A. Risk-based access control model: A systematic literature review. *Future Internet*. 2020. Vol. 12, No. 6. Art. 103. DOI: https://doi.org/10.3390/fi12060103.
6. Zadeh L. A. Fuzzy sets. *Information and Control*. 1965. Vol. 8, No. 3. P. 338–353. DOI: https://doi.org/10.1016/S0019-9958(65)90241-X.
7. Cheng P.-C., Rohatgi P., Keser C., Karger P. A., Wagner G. M., Reninger A. S. Fuzzy multi-level security: An experiment on quantified risk-adaptive access control. *Proceedings of the IEEE Symposium on Security and Privacy*. IEEE, 2007. P. 222–230. DOI: https://doi.org/10.1109/SP.2007.21.
8. Ni Q., Bertino E., Lobo J. Risk-based access control systems built on fuzzy inferences. *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*. ACM, 2010. P. 250–260. DOI: https://doi.org/10.1145/1755688.1755719.
9. Kandala S., Sandhu R., Bhamidipati V. An attribute based framework for risk-adaptive access control models. *Proceedings of the 6th International Conference on Availability, Reliability and Security*. IEEE, 2011. P. 236–241. DOI: https://doi.org/10.1109/ARES.2011.41.
10. Shaikh R. A., Adi K., Logrippo L. Dynamic risk-based decision methods for access control systems. *Computers & Security*. 2012. Vol. 31, No. 4. P. 447–464. DOI: https://doi.org/10.1016/j.cose.2012.02.006.
11. Servos D., Osborn S. L. Current research and open problems in attribute-based access control. *ACM Computing Surveys*. 2017. Vol. 49, No. 4. Art. 65. P. 1–45. DOI: https://doi.org/10.1145/3007204.
12. Riad K., Cheng J. Adaptive XACML access policies for heterogeneous distributed IoT environments. *Information Sciences*. 2021. Vol. 548. P. 135–152. DOI: https://doi.org/10.1016/j.ins.2020.09.051.
13. Kalaria R., Kayes A. S. M., Rahayu W., Pardede E., Shahraki A. S. Adaptive context-aware access control for IoT environments leveraging fog computing. *International Journal of Information Security*. 2024. Vol. 23. P. 3089–3107. DOI: https://doi.org/10.1007/s10207-024-00866-4.
14. Rose S., Borchert O., Mitchell S., Connelly S. Zero trust architecture. NIST SP 800-207. National Institute of Standards and Technology, 2020. DOI: https://doi.org/10.6028/NIST.SP.800-207.
15. Temoshok D., Fenton J. L., Choong Y.-Y., Lefkovitz N., Regenscheid A., Galluzzo R., Richer J. P. Digital identity guidelines: Authentication and authenticator management. NIST SP 800-63B-4. National Institute of Standards and Technology, 2025. DOI: https://doi.org/10.6028/NIST.SP.800-63B-4.
16. World Wide Web Consortium. Web authentication: An API for accessing public key credentials—Level 3. W3C Recommendation. 2026. August 25. URL: https://www.w3.org/TR/2026/REC-webauthn-3-20260825/.
17. Schaad J. CBOR object signing and encryption (COSE): Structures and process. RFC 9052. RFC Editor, 2022. URL: https://www.rfc-editor.org/rfc/rfc9052.html.
18. Geisler M., Pöhn D., Hommel W. Hooked: A real-world study on QR code phishing. *Sicherheit in vernetzten Systemen: 31. DFN-Konferenz*. epubli, 2024. P. B1–B18. URL: https://dpoehn.de/en/publications/.
19. Simon R. T., Zurko M. E. Separation of duty in role-based environments. *Proceedings of the 10th Computer Security Foundations Workshop*. IEEE, 1997. P. 183–194. DOI: https://doi.org/10.1109/CSFW.1997.596811.
20. European Parliament and Council of the European Union. Regulation (EU) 2016/679 (General Data Protection Regulation). *Official Journal of the European Union*. 2016. Vol. L 119. P. 1–88. URL: https://eur-lex.europa.eu/eli/reg/2016/679/oj.
21. Zanella A., Bui N., Castellani A. P., Vangelista L., Zorzi M. Internet of Things for smart cities. *IEEE Internet of Things Journal*. 2014. Vol. 1, No. 1. P. 22–32. DOI: https://doi.org/10.1109/JIOT.2014.2306328.
22. OASIS. eXtensible Access Control Markup Language (XACML) version 3.0. OASIS Standard. 2013. January 22. URL: https://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html.
23. Херсонський державний університет. KSU24. URL: https://ksu24.kspu.edu/ (дата звернення: 29.09.2026).

24. Kayes A. S. M., Rahayu W., Watters P. A., Alazab M., Dillon T., Chang E. Achieving security scalability and flexibility using fog-based context-aware access control. *Future Generation Computer Systems*. 2020. Vol. 107. P. 307–323. DOI: https://doi.org/10.1016/j.future.2020.02.001.
25. Bormann C., Hoffman P. Concise binary object representation (CBOR). RFC 8949. RFC Editor, 2020. URL: https://www.rfc-editor.org/rfc/rfc8949.html.
26. Schaad J. CBOR object signing and encryption (COSE): Initial algorithms. RFC 9053. RFC Editor, 2022. URL: https://www.rfc-editor.org/rfc/rfc9053.html.
27. Fett, D., Campbell, B., Bradley, J., Lodderstedt, T., Jones, M., & Waite, D. (2023). *OAuth 2.0 demonstrating proof of possession (DPoP)* (RFC 9449). RFC Editor. URL: https://www.rfc-editor.org/rfc/rfc9449.html
28. Bizagi (n.d.). *Import a diagram from BPMN*. Bizagi Modeler documentation. Retrieved October 31, 2026. URL: https://help.bizagi.com/platform/en/import_diagram_from_bpmn.htm
29. Bizagi (n.d.). *Simulation in Bizagi*. Bizagi Modeler documentation. Retrieved October 31, 2026. URL: https://help.bizagi.com/platform/en/simulation_in_bizagi.htm
30. Bizagi (n.d.). *Configure Simulation*. Bizagi Modeler documentation. Retrieved October 31, 2026. URL: https://help.bizagi.com/platform/en/configure_simulation.htm

Про авторів / About the authors

Олександр Лемещук, доктор філософії в галузі інформаційних технологій, проректор з цифровізації, інноваційної, інвестиційної діяльності та науково-педагогічної роботи, викладач, кафедра комп'ютерних наук та програмної інженерії, Херсонський державний університет, вул. Шевченка, 14, м. Івано-Франківськ, 76018, Україна;

Oleksandr Lemeshchuk, PhD in Information Technologies, Vice-Rector for Digitalization, Innovation, Investment, Research and Education, Lecturer, Department of Computer Science and Software Engineering, Kherson State University, 14 Shevchenka St., Ivano-Frankivsk 76018, Ukraine;

Дмитро Кльонов, викладач, аспірант, кафедра інформатики, Херсонський державний університет, вул. Шевченка, 14, м. Івано-Франківськ, 76018, Україна;

Dmitriy Klionov, Lecturer and PhD student, Department of Informatics, Kherson State University, 14 Shevchenka St., Ivano-Frankivsk 76018, Ukraine;

Денис Сенчишен, аспірант, кафедра комп'ютерних наук та програмної інженерії, Херсонський державний університет, вул. Шевченка, 14, м. Івано-Франківськ, 76018, Україна;

Denys Senchyshen, PhD student, Department of Computer Science and Software Engineering, Kherson State University, 14 Shevchenka St., Ivano-Frankivsk 76018, Ukraine;

Ростислав Засухін, здобувач другого (магістерського) рівня вищої освіти за спеціальністю «Комп'ютерні науки», кафедра комп'ютерних наук та програмної інженерії, Херсонський державний університет, вул. Шевченка, 14, м. Івано-Франківськ, 76018, Україна;

Rostyslav Zasukhin, Master's student in Computer Science, Department of Computer Science and Software Engineering, Kherson State University, 14 Shevchenka St., Ivano-Frankivsk 76018, Ukraine;

Отримано / Received 29.09.2026
Прийнято до друку / Accepted 02.11.2026
Опубліковано / Published 25.11.2026