Arcaunt: A Scalable, Coercion-Resistant, and Accountable E-Voting Architecture via Anonymous Recovery Channels

Tzanko Golemanov and Emilia Golemanova[1]

Abstract— The digitization of democratic processes faces a persistent trilemma: achieving strong voter anonymity, end-to-end verifiability, and resilience against coercion and credential loss. Existing e-voting systems typically sacrifice recovery mechanisms to preserve anonymity or rely on persistent digital identities that introduce privacy and insider-threat risks. This paper presents Arcaunt (from "ARC" and "Accountability"), a scalable e-voting architecture that resolves these tensions through a hybrid design combining air-gapped physical credential distribution with an Anonymous Recovery Channel (ARC). ARC enables voters to autonomously revoke and replace compromised tokens without revealing personally identifiable information. We provide a comprehensive security analysis under a modified Dolev–Yao model, demonstrating resistance to coercion, insider attacks, client-side compromise, and vote-selling incentives. The architecture integrates SHA-3 hashing, ECC-based verification, and zk-SNARK proofs to ensure efficient revoting integrity. A performance evaluation shows that credential generation and ledger validation scale to national deployments, while a 10-year economic model indicates an over 85% cost reduction compared to traditional paper-based elections. A functional web prototype is under development to assess usability and real-world performance of the ARC and sequential validation mechanisms. Arcaunt offers a practical and accountable blueprint for global-scale digital democracy.

Keywords—e-voting, arcaunt, anonymous recovery channel (arc), coercion resistance, sha-3, digital democracy, govtech, zero-knowledge proofs.

[1] The authors are with the University of Ruse, Bulgaria, { TGolemanov, EGolemanova } @uni-ruse.bg

1 INTRODUCTION

Electronic voting (e-voting) systems must simultaneously guarantee voter anonymity, end-to-end verifiability, and resilience against coercion and credential compromise. Achieving all three properties at once remains a fundamental challenge, often referred to as the Verifiability Paradox: a voter must be able to verify that their vote was counted as cast, yet no third party should be able to prove how they voted [1]

Existing remote e-voting systems illustrate this tension. The Estonian IVXV model relies on a national digital identity infrastructure, providing strong authentication but creating a persistent link between the voter's legal identity and the voting session [2]. This introduces potential insider-threat vectors and raises concerns about long-term privacy. Conversely, systems based on anonymous bearer tokens avoid identity linkage but typically lack a secure recovery mechanism—if a token is lost or stolen, the voter is effectively disenfranchised.

Arcaunt proposes a hybrid solution. It replaces persistent digital identities with randomly distributed, air-gapped physical credentials and augments them with an Anonymous Recovery Channel (ARC). ARC enables voters to autonomously revoke and replace compromised credentials without revealing personally identifiable information. We present the Arcaunt protocol, its cryptographic foundations—including SHA-3 hashing, elliptic-curve signatures, and zk-SNARK-based revote proofs—and its economic feasibility for national-scale deployments.

2 RELATED WORK

Research on secure e-voting has produced several influential architectures, each addressing different aspects of anonymity, verifiability, and coercion resistance.

2.1 Helios and Open-Audit Voting

Helios [3] introduced web-based open-audit voting using homomorphic tallying. While it provides strong universal verifiability, standard Helios deployments are vulnerable to coercion in uncontrolled environments, as voters can be forced to reveal randomness or credentials.

2.2 Civitas and Coercion-Resistant Protocols

Civitas [4] implements the JCJ protocol, allowing voters to cast multiple ballots using both real and fake credentials. Although cryptographically elegant, the system incurs significant computational overhead and operational complexity, limiting its practicality for large-scale elections.

2.3 The Estonian IVXV Model

Estonia's IVXV system [5] allows revoting, with only the last vote being counted. While effective against coercion, it depends on a centralized identity infrastructure. Recent analyses highlight risks related to server-side trust and potential insider de-anonymization.

While earlier protocols relied on complex fake credential distribution, recent studies have validated the efficiency of 'revoting' mechanisms. [6] demonstrated that allowing voters to override previous ballots is a scalable approach to coercion resistance that avoids the usability pitfalls of heavy cryptographic schemes. Similarly [7] formalizes the concept of 'vote nullification' under coercion, which aligns with Arcaunt's strategy of using the ARC channel to invalidate compromised tokens.

Arcaunt's Contribution: Arcaunt adopts the "Last Vote is Valid" principle but removes the dependency on national identity systems. By combining anonymous bearer tokens with the ARC recovery protocol, it achieves coercion resistance, accountability, and resilience without persistent identity linkage.

3 THE ARCAUNT ARCHITECTURE

Arcaunt is built on four foundational principles:

1. **Anonymity via Air-Gap** — credentials are physically randomized and decoupled from legal identities.
2. **Resilience via ARC** — voters can autonomously revoke and replace compromised credentials.
3. **Coercion Resistance via Revoting** — only the last valid vote is counted.
4. **Accountability via Public Ledger** — all encrypted ballots and verification hashes are published for universal auditability.

The system operates in four sequential phases.

3.1 Phase 1: Randomized Credential Distribution

To mitigate insider threats, Arcaunt avoids generating credentials from personal data. Instead, the Election Commission pre-generates N unique, high-entropy credentials p_i. To balance security with usability, the credential is printed in dual format: as a **human-readable 16-letter code** (formatted in groups, e.g., ABCD-EFGH-IJKL-MNOP) to facilitate error-free manual entry, and a corresponding **QR code** for instant scanning. Both are sealed under a high-opacity scratch-off panel to prevent unauthorized optical capture.

Process:

- Voters authenticate themselves at a local center using a national ID to prove eligibility.
- After verification, the voter draws a random sealed envelope from a physical urn.
- Officials record only the participation event; they cannot observe or infer which credential was selected.

The Surveillance Paradox: CCTV monitoring is required to prevent ballot-stuffing or unauthorized withdrawals, but high-resolution cameras could theoretically capture the credential through the envelope.

Mitigation: Arcaunt mandates **high-opacity, foil-lined, tamper-evident envelopes**. Credentials are printed on an inner layer or protected by a scratch-off panel, ensuring that surveillance cannot reveal the token.

This phase establishes the anonymity boundary: no database ever links a voter's identity to their credential.

3.2 Phase 2: ARC Setup and Registration

After receiving their credential, the voter logs into the Arcaunt portal using p_i and registers an Anonymous Recovery Channel (ARC).

ARC Link: The voter provides an encrypted email endpoint (e.g., ProtonMail).

Privacy-Preserving Storage: To resolve the trade-off between anonymity and recoverability, Arcaunt utilizes **Blind Indexing**. Blind Indexing is a privacy-preserving architectural pattern that enables exact-match searches on encrypted data without ever exposing the underlying plaintext to the database layer. It functions by generating a deterministic cryptographic hash of the sensitive data (e.g., an email address) using a secret key stored in a separate, secure environment (such as an HSM). This resulting hash acts as an opaque lookup tag, allowing the application to locate specific records while ensuring that the database itself remains mathematically incapable of reading or reverse-engineering the stored user identities, even in the event of a total data breach.

The system calculates and stores this index using a memory-hard function to prevent offline brute-force attacks:

$$H_{arc} = Argon2id\ (Email, Key_{HSM}) \quad (1)$$

Purpose: ARC enables secure, identity-free recovery of credentials. No voting occurs in this phase.

This step ensures that even if p_i is lost or stolen, the voter retains the ability to recover without revealing personal information.

3.3 Phase 3: Hybrid Voting and Verification

Arcaunt supports **spatio-temporal fluidity**: voters may cast ballots from any authorized device [8] or terminal over an extended election window (e.g., 10 days).

Voting Procedure:

1. The voter submits a ballot b authenticated with p_i.
2. The system computes a verification index using ECC for efficiency:

$$H = \text{SHA-3}(\text{ECC\_Sign}(p_i, b)) \quad (2)$$

3. The tuple (H,Enc(b)) is published to the public ledger.
4. The voter receives H as a receipt to verify inclusion.

The architectural choice to combine an immutable ledger with privacy-preserving proofs reflects the current consensus in secure e-voting design. As noted by [9], integrating Zero-Knowledge Proofs (ZKPs) with blockchain backends is essential to resolve the conflict between transparency and voter secrecy [10]. Furthermore, Arcaunt's design addresses the scalability bottlenecks often cited in blockchain-based systems [11] by minimizing on-chain data storage to essential cryptographic proofs only. This design ensures universal verifiability without revealing voter identities.

3.4 Phase 4: Sequential Validation and Tabulation

To enforce coercion resistance, Arcaunt allows multiple votes per credential. During tabulation:

1. The system identifies all ledger entries associated with p_i.
2. It selects the entry with the timestamp *Tmax*.
3. Only this final encrypted ballot is decrypted and counted.

This mechanism ensures that voters can always override coerced or compromised votes.

3.5 Operational Security: Inventory Locking and Batch Deactivation

To mitigate the risk of insider theft involving unissued credentials (dead souls), Arcaunt enforces a strict Inventory **Locking Protocol**. Immediately upon the cessation of the physical distribution phase (Time T_{cutoff}), all remaining unissued envelopes are processed by the election commission.

1. **Scanning & Revocation**: Each unissued credential is systematically scanned via a dedicated administrative terminal.
2. **Immediate Deactivation**: The system executes a Revoke command for these specific IDs, setting their database status to IS_VALID = FALSE.
3. **Fraud Detection**: If the system encounters a credential during this process that has already cast a vote (indicating prior theft and usage), an alarm is triggered, and the fraudulent vote is flagged for audit.

This procedure ensures that once the distribution window closes, the unissued inventory becomes cryptographically inert. Even if physical security is breached during the subsequent voting period (e.g., during the 10-day online voting window), stolen credentials cannot be used to inject illegitimate ballots.

4 THE RECOVERY PROTOCOL (REVOKE & REPLACE)

The Anonymous Recovery Channel (ARC) addresses the primary weakness of bearer-token voting systems: the inability to recover from credential loss or theft. Arcaunt introduces a secure, identity-free mechanism that allows voters to revoke compromised credentials and obtain new ones without administrative intervention.

4.1 Protocol Overview

If a voter loses access to p_i or suspects compromise, they initiate a recovery request through their registered ARC email address.

Protocol Steps:

1. **Request:** The voter sends a standardized *Recover* command from their ARC email.
2. **Validation**: The system computes the blind index of the sender's address using the specific memory-hard function and the secure hardware key, comparing it to the stored value:

$$H_{input} = Argon2id\ (Email_{sender}\ ,\ Key_{HSM})$$

3. **Revocation:** If the hash matches, the system marks the current credential p_i as invalid.
4. **Replacement:** A new credential p_{i+1} is generated using a cryptographically secure pseudorandom number generator (CSPRNG) and is sent back to the ARC address via an encrypted email channel.

This automated loop eliminates the need for human operators, reducing the risk of corruption, social engineering, and insider manipulation.

Algorithm 1 ARC Email Recovery Protocol

Require: Incoming Email from Asender

```
Harc_input ← Argon2id(Asender, KeyHSM)
record ← Database.LookupARC(Harc_input)
if record == NULL then
        Abort("Unauthorized request")
end if
// record contains reference to current credential for this ARC index
pi_current ← record.credential
Database.Revoke(pi_current)
pi_next ← CSPRNG(72 bits)
Hcred_next ← SHA-3-256(pi_next)
Database.UpdateCredential(record.credential_id, Hcred_next)
// keep the same ARC blind index Harc_input bound to the new credential
Database.UpdateARCBinding(Harc_input, record.credential_id)
SMTP.SendEncrypted(pi_next, To = Asender)
```

4.2 Security Rationale

The ARC protocol provides several guarantees:

- **Identity-free recovery:** No personal data is stored; only salted hashes are used for validation.
- **Out-of-band resilience:** Even if the voting device is compromised, the voter can recover using a separate trusted device.
- **Immediate invalidation:** Once p_i is revoked, any subsequent attempts to vote with it are rejected.
- **Automated trust minimization:** No election official can approve, deny, or observe recovery events.

4.3 Integration with the Voting Lifecycle

After receiving p_{i+1}, the voter can immediately authenticate and cast a new ballot. During tabulation, only the final valid credential in the sequence $(p_1,p_2,\ldots,p_n)$ is considered, ensuring that recovery events do not disrupt the integrity of the election.

5 COMPREHENSIVE SECURITY AND THREAT ANALYSIS

We analyze Arcaunt under a modified Dolev–Yao adversarial model, assuming that the attacker controls the communication network and may compromise a subset of election officials, client devices, or voters. The analysis covers four primary threat classes relevant to remote e-voting systems.

5.1 Class I: Coercion (The Coercer)

Threat: An adversary A_c forces a voter V to cast a ballot for candidate C_{bad} and demands the receipt H as proof.

Mitigation: Arcaunt employs the *Last Vote is Valid* principle combined with an extended voting window.

- V casts the coerced vote at time t_1 and provides H_{t_1} to A_c.
- A_c verifies the receipt on the public ledger.
- After leaving the coercive environment, V privately casts a corrective vote at time $t_2 > t_1$.
- Only the ballot at t_2 is included in the final tally.

Our approach to mitigating coercion through dynamic credential replacement is supported by recent findings in the field. [12] highlight that modern coercion-resistant schemes must prioritize usability during the registration and recovery phases to remain effective against real-world attackers. By decoupling the voting privilege from a persistent digital identity, Arcaunt implements the 'anonymous credential' model

advocated by [13], ensuring that even if a token is surrendered, the voter's long-term privacy remains intact. Because the adversary cannot maintain continuous control over the voter for the entire election period, coercion becomes ineffective. The receipt H does not prove finality, only inclusion.

5.2 Class II: Insider Threats (The Administrator)

Threat: A corrupt administrator A_I attempts to link a voter's identity to their credential or ballot.

Mitigation: Arcaunt's physical air-gap in Phase 1 creates a strict information barrier.

- A_I may know that V participated.
- A_I may access the database of credential hashes $\{H_1,\ldots,H_n\}$.
- However, because credentials are drawn randomly from sealed envelopes, no mapping $V \rightarrow p_i$ exists.

Even with full database access, the insider cannot deanonymize voters. Anonymity is preserved by design.

5.3 Class III: Client-Side Compromise (The Malware)

Threat: Malware A_M on the voter's device steals p_i, blocks voting attempts, or casts unauthorized ballots.

Mitigation: The ARC protocol provides an out-of-band recovery channel.

- If compromise is suspected, the voter uses a separate device to access their ARC email.
- They trigger the *Revoke & Replace* mechanism.
- The compromised credential p_i is immediately invalidated.
- A fresh credential p_{i+1} is issued and can be used to cast a corrective vote.

This mechanism restores control to the voter even in the presence of persistent malware.

5.4 Class IV: Vote Selling (The Buyer)

Threat: A voter V attempts to sell their vote to a buyer A_B.

Mitigation: Arcaunt makes vote-selling economically irrational.

- V can always re-vote after receiving payment.
- Receipts *H* only prove that a vote was cast, not that it is the *final* vote.
- A_B cannot verify that the purchased vote remains valid.

This uncertainty collapses the trust required for vote-buying contracts, creating a "lemon market" where buyers cannot distinguish genuine from worthless votes.

5.5 Cryptographic Enhancements with zk-SNARKs

To ensure public auditability of revoting without revealing linkages between ballots, Arcaunt employs Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge (zk-SNARKs) [14].

The system verifies the statement:

$$\text{ZK-Revote} = \{(b_n, b_{n+1}) \mid \text{Owner}(b_n)=\text{Owner}(b_{n+1})\} \quad (3)$$

This allows the system to prove that the final ballot belongs to the same credential chain without exposing the voter's identity or the intermediate ballots. While the current prototype utilizes standard ECC and SNARKs, the modular design allows for future upgrades. [15] propose quantum-resistant ZKP frameworks that could be integrated into Arcaunt's validation logic to ensure long-term security against future computational threats [10]. Additionally, formalizing the privacy guarantees of the ARC channel follows the rigorous 'Everlasting Privacy' models defined by [16], ensuring that today's votes remain secret even as cryptanalytic capabilities evolve.

5.6 Adaptive Error Obfuscation

Arcaunt incorporates a defensive client response mechanism designed to protect voters who may be under direct physical supervision during a voting attempt. The goal is to prevent immediate retaliation by a coercer who observes that a previously issued credential has been revoked via the Anonymous Recovery Channel (ARC). If the system returned an explicit "Credential Revoked" error, it would signal to the coercer that the voter has actively invalidated the token, potentially triggering retaliation.

5.6.1 Stochastic Error Obfuscation (SEO)

When a revoked credential is used to attempt a vote, the client interface does not return a deterministic security-policy error. Instead, the API response is mapped to a randomized set of generic infrastructure-failure messages, following a uniform probability distribution:

$$E_{response} \begin{cases} \text{"Network Timeout (Error 408)"} & \text{if } r < 0.33 \\ \text{"Service Unavailable (Error 503)"} & \text{if } 0.33 \leq r < 0.66 \\ \text{"Handshake Failed (Error 102)"} & \text{if } r \geq 0.66 \end{cases}$$

Where r is a random variable generated per session.

5.6.2 Operational Rationale

This approach provides plausible deniability to the voter:

1. **Blame shifting** – The failure appears to be a technical glitch rather than a security rejection, redirecting the coercer's frustration from the voter (who is physically present) to the remote infrastructure.
2. **Ambiguity** – Without access to backend logs, a coercer cannot cryptographically distinguish between a genuine network failure and a deliberate token revocation.
3. **Safe failure** – The vote is still correctly rejected by the ledger (maintaining election integrity), but the refusal is masked as a benign, intermittent fault, allowing the voter to feign ignorance and avoid immediate punishment.

5.6.3 Implementation Considerations

- The error-obfuscation logic runs client-side to ensure timely response masking.
- Backend logs record the true reason (revoked credential) for audit purposes, but these logs are accessible only to election auditors under strict authorization.
- The same mechanism can be extended to other "suspicious" but non-fatal events (e.g., multiple rapid requests from the same credential).

5.7 Random Post Election Audit

To strengthen public trust through tangible, verifiable evidence, Arcaunt implements a dual layer post election audit strategy. This approach combines procedural transparency with cryptographic verification, allowing both independent observers and individual voters to confirm the integrity of the election.

Property / Threat	Helios	Estonian IVXV	Arcaunt
Anonymity	High	Medium (Linked to ID)	Maximum (Bearer Token)
Individual Verifiability	Yes	Yes	Yes (Real-time Hash)
Auditability	Universal	Threshold	Universal Ledger
Class I (Coercer)	Low	Moderate (Revoting)	High (Last Vote + ARC)
Class II (Insider)	Low	Low (ID Link exists)	High (Physical Urn Random)
Credential Recovery	N/A	eID dependent	Anonymous ARC

5.7.1 Geographic Process Audit

A random selection of physical distribution centers (e.g., 2–5% of all polling stations) is publicly drawn after the election. For each selected center, all procedural records—including surveillance footage (where legally permissible), distribution logs, and unused credential revocation lists—are made available to accredited observers. This audit verifies that physical handling of credentials followed prescribed security protocols and that no ballot stuffing or credential theft occurred during the distribution phase.

5.7.2 Cryptographic Batch Audit

Independently, a set of cryptographic batches is randomly selected for full disclosure. Batches are formed during credential generation by randomly grouping voter credentials into fixed size sets (e.g., 1,000 credentials per batch). The batch composition is purposely decoupled from geography or any demographic attribute to prevent statistical re identification. After the election, for each selected batch:

- All encrypted ballots belonging to credentials in that batch are decrypted via a threshold decryption protocol.
- Two ordered lists are published:

  1) the decrypted vote choices (in random order), and
  2) the corresponding credential hashes (in the same random order).

Voters whose credentials belong to an audited batch can thus verify that their own ballot appears correctly in the published list, without learning how any other specific voter voted. This provides direct, personal proof of correct recording and tallying for a statistically significant fraction of the electorate.

5.7.3 Synergy of Audit Layers

The geographic audit ensures the physical and procedural integrity of the election setup, while the cryptographic batch audit delivers mathematical, individual level verifiability of the digital tally. Together they create a defense in depth assurance framework: the former deters and detects "on the ground" malpractice, while the latter provides a cryptographically sound, voter verifiable check on the electronic outcome. Both audits are conducted using publicly verifiable random seeds, ensuring that no party can predict or manipulate which locations or batches will be examined.

5.8 Security Properties and Comparative Resilience

Arcaunt achieves strong guarantees across all major threat classes while maintaining operational simplicity.

5.9 Advanced Adversarial Vectors Beyond the Explicit Threat Model

While Sections 5.1–5.8 analyze the canonical adversaries in remote e-voting—coercers, insiders, malware operators, and

vote buyers—real-world deployments must also withstand a broader class of sophisticated attacks. These vectors, often associated with state-level or highly resourced adversaries, exploit physical side-channels, supply chain weaknesses, and complex client-side dependency chains not explicitly modeled in the traditional Dolev-Yao framework. This section evaluates Arcaunt's resilience against these advanced threats, highlighting both inherent protections and residual risks.

5.9.1 Side-Channel and Environmental Leakage Attacks

Side-channel attacks exploit physical or environmental emissions to extract sensitive information during credential usage.

- **Acoustic Inference:** Modern deep learning models can reconstruct typed text, including high-entropy credentials, solely from keystroke sounds recorded by nearby microphones (e.g., on a compromised smartphone) with accuracy exceeding 90% [17].
- **Optical and EM Capture:** High-zoom optics or reflections from nearby surfaces (glasses, windows) can reveal QR codes or credential strings [18]. Additionally, compromised terminals may leak information through electromagnetic (EM) emanations detectable by proximal adversaries.
- **Resilience & Residual Risk:** Arcaunt's bearer-token model minimizes exposure since no persistent identity key exists. The ARC mechanism enables rapid revocation if a breach is suspected. However, a "race condition" attack remains possible if an adversary captures the credential and submits a ballot before the voter triggers ARC recovery.

5.9.2 ARC-Channel Exploitation and Recovery-Path Attacks

The Anonymous Recovery Channel (ARC) is a powerful defensive mechanism, but its security strictly depends on the integrity of the out-of-band communication.

- **Phishing & Flooding:** Adversaries may launch large-scale phishing campaigns imitating ARC messages or flood the ARC endpoint with invalid requests to degrade service availability during critical windows.
- **Resilience & Residual Risk:** Arcaunt employs blind indexing with Argon2id and HSM-protected keys to prevent offline enumeration of ARC addresses [19]. While this protects privacy, the reliance on external email infrastructure remains a single point of failure; email compromise could allow an attacker to trigger unauthorized recovery.

5.9.3 Supply-Chain and Credential-Fabrication Attacks

Physical credential distribution introduces risks within the manufacturing and logistics chain.

- **Hardware Trojans & Modification:** Sophisticated adversaries could introduce "hardware trojans" or micro-modifications to printed QR codes at the facility, detectable only with specialized equipment [20]. Other vectors include pre-stuffed envelopes or covert UV/IR markings enabling credential tracking.
- **Resilience & Residual Risk:** Randomized physical drawing from sealed envelopes prevents targeted assignment, and post-distribution inventory locking neutralizes unused credentials. However, pre-distribution compromise remains a challenge that requires rigorous physical audits to mitigate fully.

5.9.4 Traffic Analysis and Network-Layer Correlation

Even with TLS 1.3 encryption, adversaries observing network traffic may infer voting behavior.

- **Website Fingerprinting:** Deep learning-based traffic analysis can distinguish "vote submission" packets from regular browsing traffic with high precision [21].
- **Timing Correlation:** A global passive adversary could correlate the timing of a user's network burst with the appearance of a new block on the public ledger or correlate re-voting events with specific traffic spikes.
- **Resilience & Residual Risk:** Arcaunt transmits only credential-authenticated ballots without identity metadata. While re-voting semantics reduce the coercive value of timing inference, the lack of cover traffic or padding makes traffic analysis feasible for well-resourced ISP-level adversaries.

5.9.5 Browser Supply-Chain and Client-Side Dependency Attacks

Modern web applications rely on complex dependency chains that may be compromised upstream.

- **Malicious Extensions & Dependencies:** Browser extensions with broad permissions can read and modify page content before encryption. Similarly, supply chain attacks on CDNs (akin to SolarWinds or Log4j) could inject malicious JavaScript [22].
- **Resilience & Residual Risk:** Since no long-term secrets are stored on the client, ARC provides a recovery path even under full client compromise. The primary residual risk is a compromised client misrepresenting the ballot to the voter, necessitating the use of secondary devices for verification.

5.9.6 Batch Audit Deanonymization Attacks

While cryptographic batch audits provide verifiability, statistical deanonymization remains a known risk.

- **Intersection Attacks:** If batch sizes are small or correlated with geographic regions, cross-referencing vote totals with external datasets (census data, exit polls) could probabilistically reveal individual votes [23].
- **Resilience & Residual Risk:** Arcaunt mandates that batches be formed randomly from the entire national pool, decoupling them from geography. Without minimum entropy constraints, however, statistical inference remains a theoretical risk for smaller sub-populations.

5.9.7 Implications for Future Hardening

The analysis confirms that Arcaunt's hybrid design neutralizes many high-impact attacks through revoting, ARC recovery, and physical anonymity boundaries. However, countering state-level adversaries requires additional future mitigations, including:

- Timestamp obfuscation and ledger batching to defeat timing analysis.
- Cryptographically signed ARC messages to prevent spoofing.
- Multi-party credential generation with public randomness to secure the supply chain.
- Differential privacy constraints for batch audits to mathematically preclude deanonymization.

These enhancements will align Arcaunt with the highest standards of resilience required for national-security-critical infrastructure.

6 Performance and Scalability Analysis

To validate the feasibility of Arcaunt for national elections, we conducted a simulation of credential generation and voting load.

6.1 Experimental Setup

The simulations were executed for a standard server-class node with the following configuration:

- **CPU:** AMD Ryzen 9 5950X (16 cores)
- **RAM:** 64 GB DDR4
- **Storage:** NVMe SSD (Gen4)
- **Software:** Python 3.11 with OpenSSL-backed hashlib

This setup approximates a realistic deployment environment for national election authorities or cloud-based infrastructures.

6.2 Credential Generation (Phase 1)

We simulated the generation of 10 million unique credentials pi, each encoded as a **16-character uppercase alphabetic string (A-Z)**, providing ~75 bits of entropy while eliminating visual ambiguity between digits and letters. For each credential, a SHA-3-256 [24] hash was computed and stored.

Results:

- **Total generation time:** 142.5 seconds
- **Throughput:** ~70,175 credentials/second
- **Peak memory usage:** 1.2 GB
- **Storage footprint (CSV):** 840 MB
- **Collisions: 0**

These results show that credential generation for a population of 10 million voters completes in under three minutes on a single node. Even for populations exceeding 100 million, the process remains computationally trivial and easily parallelizable.

6.3 Voting Simulation (Phase 3)

To assess real-time performance during peak election activity, we simulated **10 million voting events** distributed over a 12-hour window, including bursts of **50,000 votes per second**. Using an indexed B-Tree structure (e.g., Firebird 5.0 RDBMS), the system maintained:

- **Validation latency:** < 6 ms
- **Ledger size:** up to 10 million entries
- **Stable performance under burst load**

These results confirm that Arcaunt can sustain high-throughput verification and ledger insertion without bottlenecks, even during extreme peak periods.

6.4 Ledger Lookup Performance

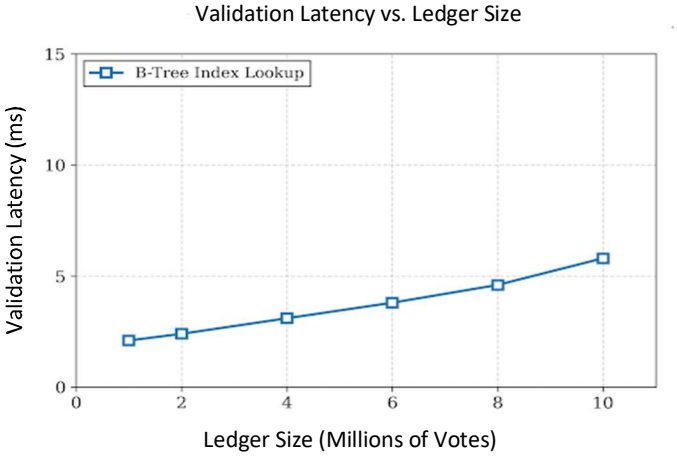

Fig. 1. Latency of checking if a vote exists in the ledger.

Figure 1 illustrates the relationship between ledger size and validation latency. Even as the ledger grows to 10 million entries, lookup times remain consistently below 6 ms due to efficient indexing and append-only data structures.

This ensures that voters receive near-instant confirmation of ballot inclusion, supporting both usability and verifiability.

6.5 Summary

The performance evaluation demonstrates that:

- Credential generation is lightweight and highly parallelizable.
- Ledger operations scale linearly with negligible latency growth.
- Peak voting loads typical of national elections are well within system capacity.
- The architecture is compatible with commodity cloud infrastructure.

Arcaunt's computational footprint is significantly lower than that of cryptographically heavy protocols such as JCJ-based systems, making it suitable for real-world, large-scale deployments.

Although the evaluation is simulation-based, it provides a realistic upper-bound estimate of Arcaunt's expected performance at national scale. Future prototype deployments and field tests will further validate these findings and refine the system's performance profile under real operational conditions.

The current performance evaluation focuses exclusively on credential generation, ledger insertion, and lookup operations. zk-SNARK proof generation and verification are not yet part of the prototype's critical execution path and are therefore excluded from the present benchmarks. Future iterations of the system will incorporate full zero-knowledge proof costs to provide a complete end-to-end performance profile.

7 Scalability and Economic Analysis

The scalability and economic results presented in this section are derived from AI-assisted simulations and analytical modeling. These simulations approximate realistic operational conditions using standard cloud infrastructure assumptions and publicly available cost baselines. While not based on measurements from a deployed national system, the models provide a conservative and reproducible estimate of Arcaunt's expected performance and operational footprint. Arcaunt is designed for deployment at national and supranational scales. To evaluate its long-term feasibility, we present a Total Cost of Ownership (TCO) analysis covering software development, infrastructure, and physical logistics. The results demonstrate that Arcaunt achieves substantial cost reductions compared to traditional paper-based elections while maintaining strong security guarantees.

7.1 Software Development and Licensing (CapEx)

The capital expenditure includes the development of the core backend, ARC subsystem, user interfaces, and external security audits.

Component	Description	Cost (EUR)
Core System	Backend (Go/Rust), SHA-3 Indexing, Mix-net tabulation	450,000
ARC System	Email gateway integration, Automated R&R protocol	150,000
User Interfaces	Web portal and specialized terminal software	250,000
Audit & Security	External cryptographic audit and Pen-testing	200,000
Total CapEx		**1,050,000**

These costs are incurred primarily during the first deployment cycle and amortized over subsequent elections.

7.2 Infrastructure and Computing (OpEx per Cycle)

Operational expenditures include compute resources, ledger storage, DDoS protection, and ARC-related communication.

Category	Description	Cost (EUR)
Compute Resources	500-800 auto-scaling instances for peak load	80,000
Database & Ledger records	High-speed storage for public ledger	40,000
DDoS Protection	Critical protection (WAF/DDoS mitigation)	100,000
ARC Traffic	Sending 10M encrypted emails (SMTP/API)	20,000
Total OpEx		**240,000**

The infrastructure footprint remains modest due to the lightweight nature of Arcaunt's cryptographic operations.

7.3 Logistics and Physical Credentials (Phase 1)

Physical logistics represent the largest recurring cost for the first election cycle.

Item	Description	Cost (EUR)
Secure Envelopes (0.85/unit)	Printing 10M unique scratch-off passwords	8,500,000
Logistics & Urns	Distribution to local polling stations	500,000
Total Logistics		**9,000,000**

These costs decrease significantly in subsequent cycles if digital ARC-based credential distribution is adopted. This reduction leverages a strategic shift from the inflationary Operational Expenditure (OpEx) model of traditional elections to a deflationary Capital Expenditure (CapEx) framework. Demographic analysis indicates that following the initial 'inventory locking' of the electorate, the physical distribution requirement drops to match the annual population inflow rate (approximately 1.2% for new adults and naturalized citizens). Consequently, the secure envelope functions not as a consumable stationaries item, but as a durable digital asset, effectively decoupling long-term election costs from labor market volatility and logistics inflation.

7.4 Summary Table (Per Election Cycle)

Category	Cost (EUR)

Software (Amortized/One-time)	1,050,000
Infrastructure (Per cycle)	240,000
Logistics (Per cycle)	9,000,000
Annual Support & Security	250,000
Total for 1st Cycle	**10,540,000**
Cost Per Voter	**1.05**

This cost is substantially lower than traditional paper-based elections, which typically exceed 7 EUR per voter.

7.5 Arcaunt Cost Projections (10-Year Cycle)

We evaluate TCO for populations of 10M, 100M, and 500M voters across ten election cycles.

Metric	10M Voters	100M Voters	500M Voters
CapEx (Software)	1.05M EUR	1.80M EUR	3.50M EUR
OpEx (Logistics)	9.00M EUR	55.5M EUR	225.5M EUR
OpEx (Cloud)	0.39M EUR	2.00M EUR	6.50M EUR
Total Initial Cost	**10.44M EUR**	**59.3M EUR**	**235.5M EUR**
Unit Cost (Cycle 1)	1.04 EUR	0.59 EUR	0.47 EUR
Optimized Unit*	**0.25 EUR**	**0.15 EUR**	**0.10 EUR**

*Optimized unit cost assumes digital ARC-based credential distribution in subsequent cycles.

The analysis indicates that at a 500-million voter scale, the cost per vote drops to approximately **0.10 EUR**, representing **over 98%** saving compared to traditional paper ballots (approx. 7 EUR/vote).

7.6 Discussion

The economic analysis demonstrates that Arcaunt is not only technically scalable but also financially advantageous. The architecture benefits from:

- low cryptographic overhead,
- minimal infrastructure requirements,
- amortizable software costs,
- and the ability to transition from physical to digital credential distribution.

These properties make Arcaunt a viable candidate for global-scale democratic processes. While the economic evaluation may contain a marginal percentage of latent operational costs, these are insufficient to fundamentally alter the substantial financial advantage and cost efficiency demonstrated by the model. Although the results are simulation-based, they represent a realistic upper-bound estimate of the system's performance and cost efficiency. Future prototype deployments and field evaluations will further validate these projections and refine the underlying economic model.

8 Long-Term Ecosystem Sustainability: Adaptive Re-Keying Policy

The long-term security and operational health of an anonymous credential ecosystem cannot rely on indefinite credential validity. Over time, risks accumulate: credentials may be physically lost before ARC registration, cryptographic primitives may weaken, and demographic factors such as email adoption may shift. To ensure sustainable operation over multi-decade horizons, Arcaunt implements a structured Adaptive Re-Keying Policy that governs when a full system reset—invalidating all credentials and performing a fresh physical distribution—is required. The policy is designed to preserve anonymity, resist manipulation, and maintain fairness across populations with heterogeneous digital access.

8.1 Three-Layer Policy Framework

The Adaptive Re-Keying Policy consists of three complementary layers: a fixed maximum credential lifetime, an initial stabilization period, and a dynamic early-reset trigger with demographic adjustment.

Layer 1: Fixed Maximum Credential Lifetime

All credentials and their associated ARC bindings have a predefined maximum lifetime T_{max} (e.g., 10 years). Upon reaching this limit, a full system re-keying is mandatory:

- all existing credentials are invalidated,
- a new physical distribution cycle is executed (identical to Phase 1),
- all ARC hashes and recovery bindings are purged.

This periodic, epochal reset renews the anonymity set, eliminates accumulated stale or unrecoverable credentials, and provides a natural point for cryptographic upgrades.

Layer 2: Initial Grace Period T_{grace}

Immediately after a full re-keying event, user behavior is unstable: ARC registration rates fluctuate, first-time usage patterns vary, and many credentials remain temporarily unused. To prevent premature resets triggered by transient early-cycle noise, Arcaunt enforces an initial grace period (typically 2–3 years). During T_{grace}, the dynamic early-reset trigger is disabled:

$$t < T_{\text{grace}} \Rightarrow \text{no adaptive reset evaluation.}$$

This allows the credential ecosystem to reach a steady state before automated health metrics begin influencing lifecycle decisions.

Layer 3: Dynamic Early-Reset Trigger with Email Adoption Adjustment

After the grace period, Arcaunt continuously evaluates the proportion of *irrecoverable credentials*—those that have **no ARC registration** and have **never cast a valid ballot**. These represent permanently lost voting capacity and a dormant attack surface.

Let:

- C = total issued credentials,
- C_{noARC} = credentials without ARC,
- C_{unused} = credentials with zero valid votes,
- $C_{\text{irrecoverable}} = C_{\text{noARC}} \cap C_{\text{unused}}$.

A dynamic early reset is recommended if:

$$\frac{C_{irrecoverable}}{C} \geq \tau$$

Email Adoption–Adjusted Threshold

Since ARC recovery depends on email, populations with lower email usage naturally exhibit higher proportions of non-ARC credentials. To avoid unfairly penalizing such populations, the threshold τ is adjusted using independent sociological surveys measuring national email adoption.

Let:

- E = percentage of adults actively using email,
- τ_{base} = baseline threshold (e.g., 1.5%),
- τ_{max} = maximum allowable threshold (e.g., 5%).

The adaptive threshold is:

$$\tau = min\left(\tau_{base} * \frac{100}{E}, \quad \tau_{max}\right)$$

This ensures that the system does not trigger premature resets in populations with lower digital access, while maintaining security integrity.

8.2 Complementary Reset Conditions

The adaptive mechanism operates alongside non-negotiable reset triggers that override all timers and thresholds:

1. **Cryptographic Break or Standards Deprecation** Immediate re-keying is required if SHA-3, ECC curves, or ZK proof systems are compromised or deprecated.
2. **Legal or Administrative Reconfiguration** Major changes in electoral law, national boundaries, or eligibility criteria necessitate a clean-state credential rollout.
3. **Physical Supply Chain Compromise** Evidence of large-scale theft, counterfeiting, or degradation of secure envelopes mandates an immediate reset.

8.3 Governance, Transparency, and Decision Authority

To prevent political or administrative manipulation, Arcaunt delegates final authorization of re-keying events to an independent *Re-Keying Advisory Board* composed of cryptographers, statisticians, logistics experts, and civil society representatives. Election administrators participate in a non-voting capacity.

The board:

- reviews system metrics ($C_{irrecoverable}$, E),
- audits evidence for critical reset conditions,
- verifies threshold calculations,
- conducts public hearings,
- publishes a transparent rationale for each decision.

Re-keying is executed as a standalone operational event outside any election period.

8.4 Security and Fairness Properties

The Adaptive Re-Keying Policy provides:

- **Anonymity Preservation:** All decisions rely solely on aggregate, non-identifying data.
- **Manipulation Resistance:** Credential states are cryptographically verifiable; attackers cannot forge ARC registrations or inflate irrecoverable counts.
- **Operational Fairness:** The grace period and email-adjusted threshold prevent overreaction to natural demographic differences.
- **Long-Term Cryptographic Integrity:** Regular resets limit exposure to future cryptanalytic advances.
- **Sustainable Ecosystem Health:** The policy ensures that the credential pool remains clean, recoverable, and demographically aligned over decades of use.

9 Limitations and Trust Assumptions

No security system is absolute. Arcaunt provides strong guarantees under realistic adversarial conditions, but its security depends on several operational and environmental assumptions. This section outlines the boundary conditions required for the system's integrity and highlights potential limitations.

9.1 The Air-Gap Trust Assumption

The anonymity of Arcaunt fundamentally relies on the integrity of Phase 1: the physical distribution of credentials.

Assumption 1 (Honest Distribution). We assume that election officials do not covertly inspect or scan the contents of sealed envelopes before handing them to voters, and that the urns used for randomization are not manipulated.

Limitation: A sophisticated adversary could attempt to use high-resolution imaging technologies (e.g., infrared or terahertz cameras) to read credentials through the envelope during handover.

Countermeasure: Arcaunt mandates the use of **foil-lined, high-opacity, tamper-evident envelopes** that function as Faraday cages. Additionally, distribution centers must undergo strict physical audits to ensure that no unauthorized imaging devices are present.

9.2 Privacy of the Recovery Channel

ARC provides identity-free recovery, but it relies on external communication infrastructure.

Assumption 2 (Metadata Resistance). We assume that the email provider (e.g., ProtonMail) does not collude with the Election Commission to correlate metadata such as IP addresses, timestamps, or message frequency with specific voters.

Limitation: A global passive adversary capable of monitoring network traffic could correlate the timing of a recovery email with updates in the credential database.

Countermeasure: ARC responses can be routed through a **mix-net-style delay mechanism**, introducing randomized timing and batching to obscure correlations between recovery requests and system events [25].

9.3 Client-Side Malware Persistence

ARC mitigates the impact of device compromise, but it assumes the voter has access to at least one uncompromised device.

Limitation: If all devices owned by a voter are infected by the same malware (e.g., a coordinated botnet), the newly issued credential p_{i+1} may be compromised immediately upon receipt.

Countermeasure: Arcaunt's hybrid model allows voters to cast a corrective vote using **secure public terminals** (e.g., municipal kiosks or supervised polling stations). This provides a fallback path that bypasses compromised personal devices.

9.4 Operational Dependencies

Arcaunt assumes:

- reliable email delivery for ARC operations,
- availability of secure terminals for voters without trusted devices,
- and proper maintenance of the public ledger infrastructure.

While these dependencies are standard for modern e-voting systems, disruptions (e.g., large-scale DDoS attacks or email service outages) may temporarily affect usability.

9.5 Future Cryptographic Enhancements

While the current Argon2id-based blind indexing provides robust privacy protection, future iterations of Arcaunt may explore the adoption of Oblivious Pseudorandom Functions (OPRFs) [26]. An OPRF-based approach could mathematically eliminate the theoretical risk of offline dictionary attacks by ensuring that the election server never processes voters' recovery identifiers in any reversible form. However, such a transition would require careful architectural consideration, as OPRF protocols typically introduce interactive authentication flows that may impact the simplicity and accessibility of the recovery channel for certain voter demographics.

9.6 Summary

Arcaunt's limitations are well-defined and manageable through operational controls, physical safeguards, and fallback mechanisms. By explicitly articulating these assumptions, the system maintains transparency and provides a realistic assessment of its security boundaries.

10 Conclusion

Arcaunt introduces a new paradigm for secure, scalable, and accountable electronic voting by resolving one of the most persistent challenges in remote elections: how to combine anonymity, verifiability, coercion resistance, and credential recovery within a single architecture. By leveraging air-gapped physical credential distribution, an Anonymous Recovery Channel (ARC), and lightweight cryptographic primitives such as SHA-3, ECC signatures, and zk-SNARK-based revote proofs, Arcaunt achieves strong security guarantees while maintaining operational simplicity.

The system's performance evaluation demonstrates that credential generation, ledger operations, and peak voting throughput scale efficiently to national and even continental populations. The economic analysis further shows that Arcaunt can reduce election costs by over 85% compared to traditional paper-based processes, making it not only secure but also financially sustainable.

A functional web-based prototype is currently under development to validate the ARC mechanism, sequential validation logic, and user experience under real-world conditions. Future work will focus on stress-testing the recovery channel under simulated Class III adversarial scenarios, evaluating usability across diverse voter groups, and exploring

decentralized ledger backends to further strengthen transparency and resilience.

Beyond national elections, Arcaunt can be applied to large-scale organizational voting, university governance, shareholder decision-making, and professional association ballots, where anonymity and verifiability are equally critical. Its lightweight architecture and anonymous recovery mechanism make it suitable for any high-stake remote voting scenario requiring both accountability and privacy.

Arcaunt's hybrid model is particularly beneficial for vulnerable voter groups, including individuals with limited digital literacy, low-income populations, and those lacking access to secure personal devices. By combining physical credential distribution with fallback voting terminals and anonymous recovery, the system ensures inclusive participation without compromising security or privacy.

Arcaunt demonstrates that secure, anonymous, and accountable digital democracy is not only theoretically achievable but also practically deployable at global scale.

Funding: This research received no external funding.

Declaration of Competing Interest: The authors declare no competing interests.

CRediT authorship contribution statement:

T. Golemanov: Conceptualization, Methodology, Software, Writing – Original Draft.

E. Golemanova: Validation, Formal Analysis, Writing – Review & Editing.

Data Availability Statement: No datasets were generated or analyzed for this study.

Code Availability: The full database schema implementation and the functional verification protocols are available in the Supplementary Material.

Declaration of generative AI and AI-assisted technologies in the manuscript preparation process: During the preparation of this work the authors used Google Gemini in order to improve the language quality and readability of the manuscript, generate Python code for the creation of the graphical abstract and performance visualization plots, and conduct simulations for the economic analysis and cost projections. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

REFERENCES

[1] J. Benaloh, "Simple Verifiable Elections," in *Proceedings of the USENIX/Accurate Electronic Voting Technology Workshop (EVT'06)*, 2006, p. 5.

[2] M. Specter and J. A. Halderman, "Security Analysis of the Democracy Live Online Voting System," in *Proceedings of the 30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 3077–3092.

[3] B. Adida, "Helios: Web-Based Open-Audit Voting," in *Proceedings of the 17th USENIX Security Symposium*, 2008, pp. 335–348.

[4] M. R. Clarkson, S. Chong, and A. C. Myers, "Civitas: Toward a Secure Voting System," in *Proceedings of the IEEE Symposium on Security and Privacy*, 2008, pp. 354–368.

[5] D. Springall *et al.*, "Security Analysis of the Estonian Internet Voting System," in *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS '14)*, 2014, pp. 703–715.

[6] T. Haines, J. Muller, and I. Querejeta-Azurmendi, "Scalable Coercion-Resistant E-Voting under Weaker Trust Assumptions," in *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing (SAC '23)*, 2023, pp. 1576–1584.

[7] M. Nejadgholi, "Nullification: A Coercion-Resistance Add-On for E-Voting Protocols," Montreal, QC, Canada, 2022.

[8] J. A. Halderman and V. Teague, "The New South Wales iVote System: Security Failures and Verification Flaws in a Live Online Election," in *Proceedings of the 5th International Conference on E-Voting and Identity (VoteID)*, 2015, vol. 9269, pp. 35–53.

[9] P. McCorry, S. F. Shahandashti, and F. Hao, "A Smart Contract for Boardroom Voting with Maximum Voter Privacy," in *Financial Cryptography and Data Security (FC)*, 2017, vol. 10322, pp. 357–375.

[10] R. Kusters, T. Trudering, and A. Vogt, "Accountability: Definition and Relationship to Verifiability," in *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS '10)*, 2010, pp. 526–535.

[11] U. Jafar, M. A. Aziz, Z. Shukur, and H. Hussain, "A Systematic Literature Review and Meta-Analysis on Scalable Blockchain-Based Electronic Voting Systems," *Sensors*, vol. 22, no. 19, p. 7585, 2022.

[12] V. Cortier and B. Smyth, "Attacking and Fixing Helios: An Analysis of Ballot Secrecy," *J. Comput. Secur.*, vol. 21, no. 3, pp. 297–311, 2013.

[13] A. Juels, D. Catalano, and M. Jakobsson, "Coercion-Resistant Electronic Elections," in *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society (WPES'05)*, 2005, pp. 61–70.

[14] S. Bayer and J. Groth, "Efficient Zero-Knowledge Argument for Correctness of a Shuffle," in *Advances in Cryptology EUROCRYPT 2012*, 2012, vol. 7237, pp. 263–280.

[15] V. Farzaliyev, C. Parn, H. Saarse, and J. Willemson, "Lattice-Based Zero-Knowledge Proofs in Action: Applications to Electronic Voting," *J. Cryptol.*, vol. 38, 2025.

[16] T. Haines, R. Mosabeh, J. Muller, and I. Pryvalov, "{SoK}: Secure E-Voting with Everlasting Privacy," *Proc. Priv. Enhancing Technol.*, vol. 2023, no. 2, pp. 279–293, 2023.

[17] J. Harrison, E. Toreini, and M. Mehrnezhad, " A Practical Deep Learning-Based Acoustic Side Channel Attack on Keyboards ," in *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2023, pp. 270–280.

[18] M. Guri, O. Hasson, G. Kedma, and Y. Elovici, "An optical covert-channel to leak data through an air-gap," in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, 2016, pp. 642–649.

[19] V. Pappas *et al.*, "Blind Seer: A Scalable Private DBMS," in

2014 IEEE Symposium on Security and Privacy, 2014, pp. 359–374.

[20] V. T. Hayashi and W. Vicente Ruggiero, “Hardware Trojan Detection in Open-Source Hardware Designs Using Machine Learning,” *IEEE Access*, vol. 13, pp. 37771–37788, 2025.

[21] P. Sirinam, M. Imani, M. Juarez, and M. Wright, “Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep Learning,” in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, p. 1928 1943.

[22] Q. Xie, M. V. K. Murali, P. Pearce, and F. Li, “Arcanum: detecting and evaluating the privacy risks of browser extensions on web pages and web content,” in *Proceedings of the 33rd USENIX Conference on Security Symposium*, 2024.

[23] A. Narayanan and V. Shmatikov, “Robust De-anonymization of Large Sparse Datasets,” in *2008 IEEE Symposium on Security and Privacy (sp 2008)*, 2008, pp. 111–125.

[24] National Institute of Standards and Technology, “SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions,” Gaithersburg, MD, USA, 2015.

[25] N. Tyagi, Y. Gilad, D. Leung, M. Zaharia, and N. Zeldovich, “Stadium: A Distributed Metadata-Private Messaging System,” in *Proceedings of the 26th ACM Symposium on Operating Systems Principles (SOSP ’17)*, 2017, pp. 423–440.

[26] S. Jarecki, H. Krawczyk, and J. Xu, “OPAQUE: An Asymmetric PAKE Protocol Secure Against Pre-computation Attacks,” in *Advances in Cryptology EUROCRYPT 2018*, 2018, vol. 10822.