

UNIVERSITY OF MIAMI
COLLEGE of ENGINEERING

Department of Electrical and Computer Engineering

SENIOR DESIGN PROJECT

AUTHENTICATED RANDOMNESS INFRASTRUCTURE SERVICE (ARIS)

By

Maxwell Cosenza, Software Engineering
Joel Eyison, Electrical Engineering
Johnathon L'Amoreaux, Computer Engineering

Final Report

Spring 2026

Advisor: Dr. Stephen Murrell & Sakhrat Khizroev

4/28/2026

ABSTRACT

ARIS converts physical entropy from a Magneto-Optic Kerr Effect (MOKE) opticalsetup into cryptographically signed, statistically validated random data servedvia REST API. A helium-neon laser directed at a magnetic thin film near itscoercive point generates true entropy from quantum magnetic domain switching.The analog signal is extracted by an SR865A lock-in amplifier, debiased via VonNeumann, expanded into 256-bit blocks via SHA-256, and screened against NIST SP800-22 monobit and runs tests. Each accepted block is Ed25519-signed andappended to a hash-chained audit log, forming a tamper-evident chain of 11,151blocks. A REST API dispenses blocks to three demonstrations: an authenticatedblackjack game, a Monte Carlo portfolio risk simulation, and a fair auditsampling tool. All 15 NIST SP 800-22 tests pass on the combined 2.8-million-bitdataset. An independent verifier confirms signature validity, chain integrity,and timestamp precedence using only the public key.

Table of Contents

1. Introduction 1
   1.1. Problem statement 1
   1.2. Problem Description 1
   1.3. Market Needs 1
   1.4. Engineering objectives and system specifications 2
   1.5. Functional description of system sections 2
      1.5.1. Level 0 *2*
      1.5.2. Level 1 *3*
      1.5.3. Level 2 *3*
2. LITERATURE RESEARCH 5
   2.1. Background: Why Randomness Matters 5
   2.2. Survey of Existing TRNG Technologies 5
   2.3. The Magneto-Optic Kerr Effect as an Entropy Source 6
   2.4. Statistical Validation: NIST SP 800-22 6
   2.5. Cryptographic Authentication and Hash Chaining 6
   2.6. The Commit-then-Reveal Protocol 7
   2.7. Gaps Addressed by ARIS 7
3. DESIGN PROCEDURE 8
   3.1. Design procedure and team effort 8
   3.2. Design alternatives 9
   3.3. Analysis: Equations, simulations, general circuits or functional diagrams 10
4. DESIGN DETAILS 11
   4.1. Detailed design 11
      4.1.1. Hardware Entropy Subsystem 11
      4.1.2. Optical Alignment and Mechanical Stabilization 12
      4.1.3. Component Selection 13
      4.1.4. Software Processing Pipeline 13
      4.1.5. REST API 14
      4.1.6. Demo Applications 14
      4.1.7. Independent Verifier 14
   4.2. Schematics 15
5. DESIGN VERIFICATION 19
   5.1. Testing procedures 20
   5.2. Quantitative results, graphs, measurements 20
   5.3. Discussion of Results 22
6. SUSTAINABILITY, REALISTIC CONSTRAINTS, ETHICAL ISSUES, AND ENGINEERING STANDARDS 22
   6.1. United Nations-Sustainable Development Goals 23

6.2. Constraints 23
6.3. Ethical issues 23
6.4. Engineering Standards 24
6.5. Cost – Parts 25
6.6. Cost – Labor 25
7. TEAM MEMBER CONTRIBUTIONS 26
7.1. Maxwell Cosenza, Software Engineering 27
7.2. Joel Eyison, Electrical Engineering 27
7.3. Johnathon L'Amoreaux, Computer Engineering 27
8. CONCLUSIONS 29
8.1. Summary of Design Goals 29
8.2. Proposed Versus Attained Functionality 29
8.3. Future Extensions 29
Appendix 1 30

List of Figures

Figure 1.0 ARIS Level 0 block diagram 2
Figure 1.1 ARIS Level 1 architecture diagram 3
Figure 1.2 ARIS Level 2 software pipeline diagram 4
Figure 3.0 - Proposed Gantt chart 8
Figure 3.1 - Final Gantt chart 8
Figure 3.2 - ARIS full pipeline diagram 10
Figure 3.3 - Commit-then-reveal protocol timeline diagram 10
Figure 4.0 - Hardware optical path schematic 11
Figure 4.1 - MOKE optical setup photograph 15
Figure 4.2 - SR865A lock-in amplifier display 15
Figure 4.3 - ARIS system architecture overview 16
Figure 4.4 - Blockchain and cryptography diagram 17
Figure 4.5 - API dispensing diagram 18
Figure 4.6 - Verification flow diagram 18
Figure 5.0 - NIST monobit p-value histogram 21
Figure 5.1 - NIST runs test p-value histogram 21

List of Tables

Table 6.1 ARIS Optical and Detection Hardware Cost Breakdown **25**
Table 6.2 Summary of estimated labor cost for completing a system prototype **25**

1. Introduction

1.1. Problem statement

Randomness underpins security, finance, and fair computation, yet no widely deployed infrastructure exists to produce, distribute, and publicly verify physically generated random data.

1.2. Problem Description

Modern cryptographic and secure computing systems depend on randomness for keygeneration, data protection, and fair computation. Most software today usespseudorandom number generators (PRNGs) that rely on deterministic algorithms andseed values to simulate randomness. Once the seed or algorithm is known, theentire sequence can be reproduced or predicted (Barak, Shaltiel, & Tromer, n.d.).This predictability creates a single point of failure in high-assurance systems,where even small leaks or reused seeds can compromise entire encryption processesor allow tampering that appears legitimate.

The problem manifests across three major domains. In online gambling and gaming,casinos, poker rooms, and lotteries shuffle cards and generate outcomes usingsoftware PRNGs. If the seed is predictable or leaked, every past result can bereconstructed by the house, and players have no mechanism to audit the process.Fairness is claimed but never proven. In financial risk modeling, Monte Carlosimulations for Value-at-Risk, portfolio stress tests, and derivatives pricingseed their random number generators with system time or internal state. A biasedseed corrupts the output distribution, and regulators have no audit trail toindependently verify results. In cryptographic infrastructure, SSL/TLS keys,JWT tokens, API nonces, and session identifiers all depend on entropy quality.A broken entropy source silently undermines every security primitive built ontop of it.

The common failure across all three domains is that randomness operates as ablack box. There is no proof it is fair, no audit trail, and no mechanism forindependent verification.

1.3. Market Needs

MN1: Online casinos, lotteries, and gaming platforms require independentlyverifiable randomness to satisfy player trust and regulatory requirements.The global online gambling market exceeds $100 billion annually, and nowidely deployed technical standard exists for publicly auditable fairness.
MN2: Financial institutions and regulators require reproducible, auditablerandom seeds for Monte Carlo risk simulations. Under Basel III, banks musthold capital against Value-at-Risk models, but the randomness underlyingthose models is currently unverifiable.
MN3: Cryptographic infrastructure operators require high-quality entropysources whose output quality can be independently certified. Weak orpredictable entropy silently degrades every downstream security guarantee.
MN4: Auditing and sampling applications in government and accounting requirea verifiable, tamper-proof mechanism for selecting random subsets ofpopulations for review, with a permanent record that the selection was notmanipulated.
MN5: Research and academic communities require a platform that demonstrateshow physical entropy and cryptographic proof can be combined into atransparent, reproducible system with practical applications.

1.4. Engineering objectives and system specifications

ARIS is designed to meet twelve engineering requirements (ERs) across fivefunctional categories: randomness quality, throughput, security, interfaces,and safety. The complete requirements are specified in Section 3 andverified in Section 5. At a high level, the system must:

ER1 (Randomness Quality): The output data must successfully pass all tests in the NIST SP 800 22 set, using a significance level of 0.10.

ER2 (Entropy): The random data produced should have an average entropy value above 0.90 bits per bit.

ER3 (Bit Rate): The system will produce a post-processed random bitstream with a sustained throughput of at least 500 bits per second.

ER4 (Stability): The system will operate continuously for at least thirty minutes without failing ER-1 or ER-2.

ER5 (Data Interface): The FPGA will provide the generated random bitstream to a host computer through a standard serial or USB connection.

ER6 (Validation and Signing): Each random output will be verified with statistical tests and digitally signed using asymmetric cryptography.

ER7 (Logging and Audit): All signed outputs will be stored in an append-only log protected by cryptographic hash chaining.

ER8 (Verification Tool): An external verifier will be able to reproduce results from stored inputs and confirm signature and log integrity.

ER9 (API Integration): The software will expose an authenticated REST API for receiving raw entropy and delivering validated random data.

ER10 (Throughput Handling): The service will process a minimum of one thousand random-data requests per minute without data loss.

ER11 (Security and Access Control): All stored private keys, logs, and related data will be kept encrypted and only accessible to authorized users based on their roles.

ER12 (Safety): The optical setup will follow Class 3R laser safety rules, and every part that users can touch will run on less than twelve volts of direct current to keep it safe to handle.

1.5. Functional description of system sections

1.5.1. Level 0

At the highest level, ARIS is a pipeline with a single input and a singleoutput. The input is physical entropy from a MOKE optical setup. The outputis a stream of authenticated, statistically validated random blocks servedvia a REST API, each accompanied by a cryptographic proof of authenticitythat any third party can verify using only the public key.

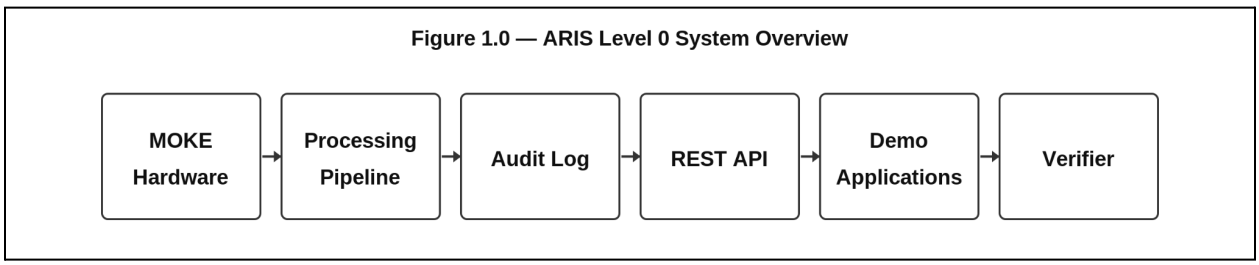

Figure 1.0 ARIS Level 0 block diagram: MOKE Hardware -> Pipeline -> Audit Log -> REST API -> Demo Applications -> Verifier

1.5.2. Level 1

At Level 1, ARIS consists of three subsystems:

Hardware Entropy Subsystem: A 632.8 nm helium-neon laser illuminates amagnetic thin film near its coercive point. Quantum magnetic domain switchingcauses random polarization changes in the reflected beam. An SR865A lock-inamplifier extracts the R and Theta signals at 10 Hz and streams them to thehost computer as a CSV file via USB-VISA.

Software Processing Subsystem: The host ingests the CSV, removes DC drift,applies Von Neumann debiasing, expands bits via SHA-256, and screens everyblock against NIST SP 800-22 monobit and runs tests. Passing blocks areEd25519-signed and appended to a hash-chained audit log.

Distribution and Verification Subsystem: A REST API (aris_api.py) dispensesblocks sequentially to demo applications. An independent verifier(verify_draw.py) confirms signature validity, hash chain integrity, andtimestamp precedence for any issued receipt.

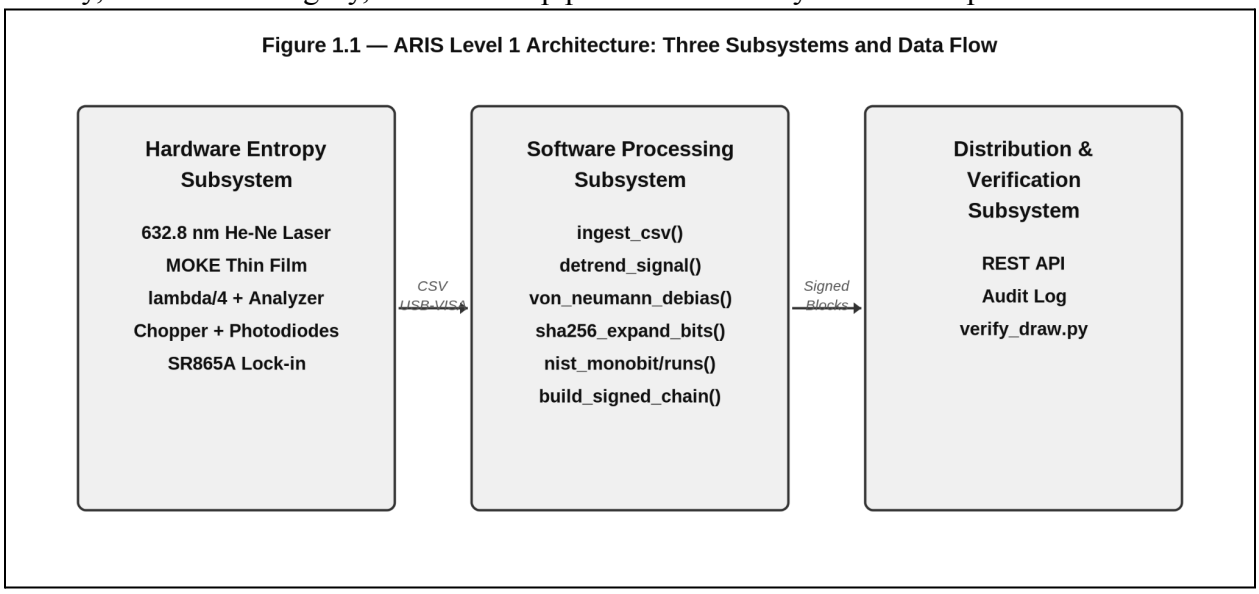

Figure 1.1 ARIS Level 1 architecture diagram showing the three subsystems and the data flow between them

1.5.3. Level 2

The hardware entropy subsystem consists of six physical stages that transform magnetic fluctuations into a clean analog signal delivered to the software pipeline:

- ➢ HeNe Laser Source (Thorlabs HNL210-series): Emits a stable, coherent 632.8 nm beam at less than 5 mW. The beam is linearly polarized and serves as the optical carrier for the MOKE measurement. Low output power keeps the setup within Class 3R safety limits.
- ➢ MOKE Thin Film: The laser passes through a magnetic thin film mounted on the optical table. The Magneto-Optic Kerr Effect rotates the plane of polarization by a small angle proportional to the instantaneous magnetization state of the film. Thermal and quantum fluctuations in the magnetic domains drive the randomness.
- ➢ Quarter-Wave Plate and Analyzer (Glan Laser Polarizer): The λ/4 plate and high-extinction Glan polarizer convert the Kerr rotation into an intensity modulation. This optical stage is the primary sensitivity control and must be precisely aligned to maximize the signal-to-noise ratio.

- ➢ Optical Chopper: A mechanical beam chopper modulates the laser at a fixed reference frequency. This reference is fed to the lock-in amplifier, allowing phase-sensitive detection that rejects all noise outside the chopper frequency band.
- ➢ Balanced Photodiodes (Thorlabs DET10A, DET36A, Hinds DET-200): Two photodiodes capture the modulated signal and a reference beam. Differential detection cancels common-mode intensity noise from the laser, isolating the Kerr signal.
- ➢ SR865A Lock-in Amplifier: Phase-sensitive detection at the chopper reference frequency extracts the R (magnitude) and Θ (phase) components of the Kerr signal from the noise floor. The SR865A outputs these as timestamped CSV rows via USB-VISA at approximately 10 samples per second, which the software pipeline ingests directly.

The software processing subsystem consists of six pipeline stages, eachimplemented as a distinct function in process_entropy.py:

- ➢ ingest_csv(): Reads Time, R, Theta columns from the SR865A CSV. Validatesschema and drops malformed rows. Approximately 3,000 samples per 300-secondcapture session.
- ➢ detrend_signal(): Applies scipy.signal.detrend() to remove slow DC drift andthermal creep from the R channel. Centers fluctuations at zero, which is aprerequisite for unbiased Von Neumann extraction.
- ➢ von_neumann_debias(): A Theta-gated extractor that compares adjacent R pairs.If R_n < R_(n+1), emit 1. If R_n > R_(n+1), emit 0. Equal pairs arediscarded. This eliminates bias regardless of the underlying probability P(1).
- ➢ sha256_expand_bits(): Seeds SHA-256 with each VN bit concatenated with its4-byte index. Produces one 256-bit block per accepted bit, dramaticallymultiplying usable output while preserving uniformity.
- ➢ nist_monobit/runs(): Applies NIST SP 800-22 inline screening to every block.Blocks failing either test at p < 0.01 are discarded before signing.
- ➢ build_signed_chain(): Signs each canonical JSON record with Ed25519. Embedsthe SHA-256 hash of the prior record as prev_record_hash, forming atamper-evident chain. Any edit to a past record cascades as a hash mismatchdetectable by verify_signed_chain().

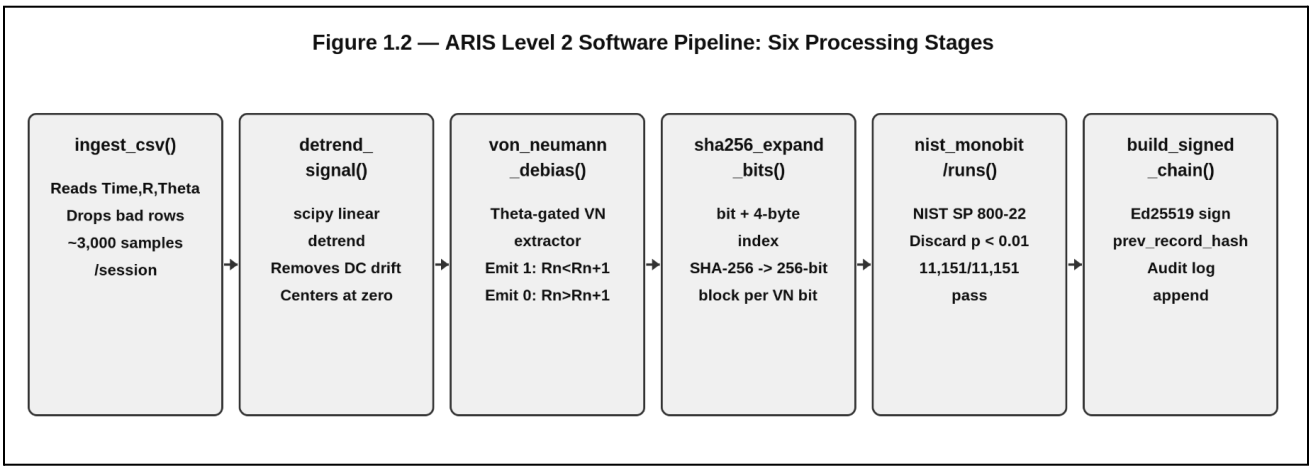

Figure 1.2 ARIS Level 2 software pipeline diagram showing the six processing stages in sequence

2. LITERATURE RESEARCH

Random number generation sits at the intersection of physics, mathematics, andcomputer security. This section surveys the relevant technical background,existing approaches, and the gaps that ARIS addresses.

2.1. Background: Why Randomness Matters

Randomness is foundational to modern cryptography. Key generation, nonceselection, session tokens, and challenge-response protocols all requireoutputs that are statistically indistinguishable from uniformly randomsequences. If the source of randomness is predictable, every cryptographicguarantee built on top of it collapses (Barak et al., n.d.). Beyond security,randomness plays a central role in scientific simulation, statistical sampling,and fairness-sensitive applications such as lotteries and audit selection.

Pseudorandom number generators (PRNGs) are deterministic algorithms thatproduce sequences with statistical properties resembling randomness, but areultimately reproducible from their seed. Widely used algorithms such asMersenne Twister and AES-CTR DRBG provide excellent statistical propertiesbut rely entirely on seed secrecy. A leaked or predictable seed allows fullreconstruction of all past and future outputs (Koç, 2017). In adversarialcontexts, this represents a fundamental vulnerability. Several real-worldexploits have demonstrated this: the Debian OpenSSL entropy vulnerabilityin 2008 resulted in predictable private keys on millions of systems, andmultiple online poker services have been compromised by reverse-engineeringtheir PRNG state from observed outputs.

True random number generators (TRNGs) address this by drawing entropy fromphysical processes that are fundamentally non-deterministic. Thermal noise,shot noise, radioactive decay, and photon path uncertainty are all examplesof physical processes that cannot, even in principle, be predicted from priorobservations. TRNGs provide a qualitatively stronger guarantee than PRNGsbecause the unpredictability is rooted in physics rather than algorithmiccomplexity (Herrero-Collantes & Garcia-Escartin, 2017)

2.2. Survey of Existing TRNG Technologies

Electronic TRNGs based on thermal or avalanche noise are widely used incommercial products. Intel's RDRAND instruction and ARM's TrustZone entropysources fall in this category. While practical and fast, these sources aresensitive to temperature variation and electromagnetic interference, and theirblack-box integration into silicon makes independent verification of entropyquality difficult. There is no external signal an auditor can examine toconfirm that the device is behaving as specified (Stipcevic, 2012).

Optical quantum TRNGs exploit the fundamental uncertainty inherent in photonbehavior. Photon path splitting at a beam splitter, photon arrival timing,and vacuum fluctuation measurements have all been demonstrated as high-qualityentropy sources. These approaches offer excellent randomness quality buttypically require laboratory-grade detectors and precise optical alignment,making them expensive and difficult to deploy outside research settings(Herrero-Collantes & Garcia-Escartin, 2017).

Magnetic domain-based entropy sources represent an emerging research direction.Work by Jiang et al. (2020) demonstrated that nanoscale magnetic skyrmiondynamics produce stochastic behavior suitable for hardware random numbergeneration. The fundamental mechanism is thermal and quantum fluctuation inmagnetic domain alignment near the coercive

point, where competing magneticstates are energetically nearly equal. Small perturbations cause irreversibledomain flips that are physically non-deterministic. ARIS applies this samephysical mechanism at the macroscopic scale via the Surface Magneto-OpticKerr Effect, which makes the domain switching optically measurable.

2.3. The Magneto-Optic Kerr Effect as an Entropy Source

The Surface Magneto-Optic Kerr Effect describes the rotation of polarizedlight upon reflection from a magnetized surface. The magnitude and directionof the rotation (the Kerr angle) depend on the orientation of the surfacemagnetization (Qiu & Bader, 2000). When a magnetic thin film is held near itscoercive point, the magnetization oscillates stochastically between competingdomain orientations driven by thermal and quantum fluctuations. This producesa time-varying Kerr signal that encodes physical randomness.

The key advantage of MOKE as a TRNG source is that the entropy is observableand measurable. Unlike a silicon TRNG whose internal state is inaccessible,the MOKE signal is an analog voltage that can be recorded, archived, andindependently analyzed. This transparency is what enables the cryptographicaudit chain that ARIS builds on top of it.

Lock-in amplification is the standard technique for extracting small periodicsignals from noisy backgrounds (Newport Corporation, 2021). By modulating theoptical beam with a mechanical chopper and synchronously demodulating thedetector output at the chopper frequency, the lock-in amplifier extracts onlythe signal component correlated with the modulation, rejecting all out-of-bandnoise. This technique is widely used in optical spectroscopy and surfacemagnetometry, and is essential for achieving a usable signal-to-noise ratio inthe MOKE entropy source.

2.4. Statistical Validation: NIST SP 800-22

The NIST Special Publication 800-22 (National Institute of Standards andTechnology, 2010) defines a suite of 15 statistical tests for evaluating therandomness of binary sequences. The tests cover frequency balance (monobit),run length distributions, spectral properties (FFT), linear complexity,serial correlation, and several others. Passing all 15 tests on a sufficientlylarge dataset (the full suite requires at minimum approximately 1 million bits)is the accepted standard for demonstrating that a sequence is statisticallyindistinguishable from uniform randomness.

ARIS applies two of the NIST tests inline on every individual 256-bit blockas a quality gate: the monobit test and the runs test. These two tests arecomputationally inexpensive and catch the most common failure modes (DC biasand autocorrelation). Full suite testing is performed on the combined datasetof all sessions. This two-tier approach ensures per-block quality whilemeeting the volume requirement for full certification.

2.5. Cryptographic Authentication and Hash Chaining

Ed25519 is an elliptic curve signature scheme based on Curve25519 offering128-bit security with small key sizes (32 bytes) and fast signing andverification (Bernstein et al., 2012). Its deterministic signing propertymeans that the same message and private key always produce the same signature,which is important for canonical JSON serialization: verify_draw.py canindependently reconstruct the signed bytes from the block record and confirmthe signature without needing to trust any intermediate software.

Hash chaining, popularized by blockchain systems, creates a tamper-evidentsequence by embedding the hash of each record in the next. ARIS applies thisat the block level: each signed record contains the SHA-256 hash of the priorrecord as prev_record_hash. Any modification to a past block changes its hash,invalidating the prev_record_hash field in every subsequent block and makingthe tampering immediately detectable by verify_signed_chain() (Lei et al., 2022).

2.6. The Commit-then-Reveal Protocol

The commit-then-reveal protocol is a cryptographic construction for provablefairness in sequential processes. The operator publicly commits to a value(in ARIS, a block ID) before the process runs, then executes the process usingthat committed value. Because the commitment predates the process, the operatorcannot have selected a favorable value after observing the outcome. Thisconstruction is used in blockchain-based fair lotteries and verifiable randomfunctions (Goldwasser et al., 2018). ARIS applies it to all three demoapplications: the block ID is announced before the blackjack deck is dealt,before the Monte Carlo simulation runs, and before the audit sample is drawn.

2.7. Gaps Addressed by ARIS

Existing TRNGs, including commercial silicon devices and academic opticalgenerators, share a common limitation: they provide no externally verifiableproof that the entropy they deliver is authentic and unmanipulated. The outputarrives as a bitstream with no provenance. ARIS addresses this gap by combininga physically observable entropy source with a cryptographic proof layer thatallows any third party to independently verify, using only the public key, thata specific block of randomness was generated at a specific time by a specificdevice, has not been altered since signing, and predates any draw or simulationthat references it.

This combination of physical transparency and cryptographic proof is novel.Prior work in TRNG literature focuses either on physical entropy quality or oncryptographic signing, but not on the end-to-end auditable pipeline that ARISimplements, including the REST API, the commit-then-reveal demo applications,and the standalone verifier.

3. DESIGN PROCEDURE

3.1. DESIGN PROCEDURE AND TEAM EFFORT

The ARIS project followed a hybrid development methodology that combinedsequential hardware development with iterative software development. Thehardware entropy subsystem required a strict sequential build order becauseoptical alignment, detector placement, and lock-in configuration are physicallydependent: each stage must be stable before the next can be tuned. Thesoftware subsystem was developed iteratively with short cycles, allowing earlyvalidation of the cryptographic pipeline using simulated entropy data beforelive hardware was available.

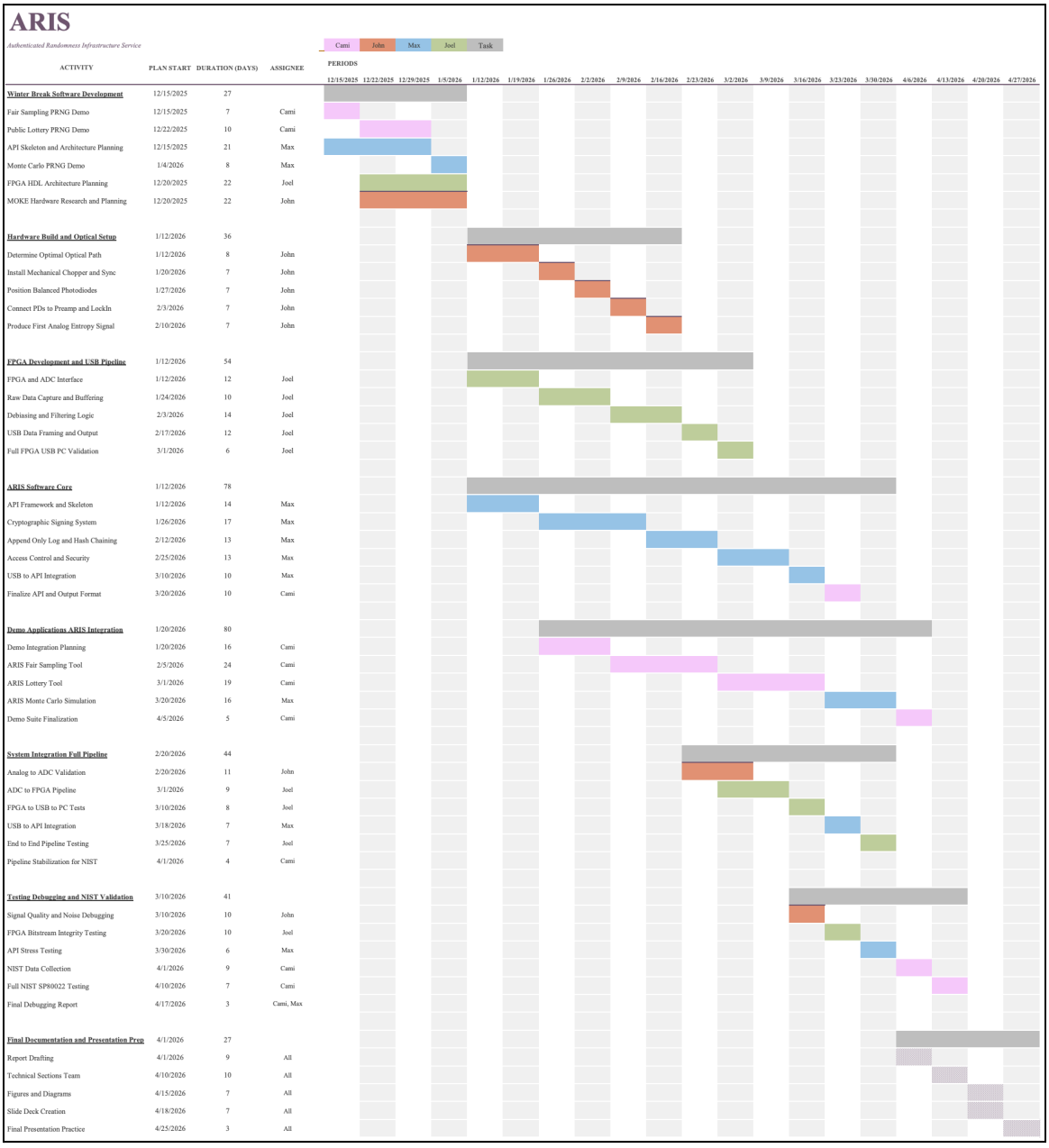

Figure 3.0 - Proposed Gantt chart from Senior Design I proposal

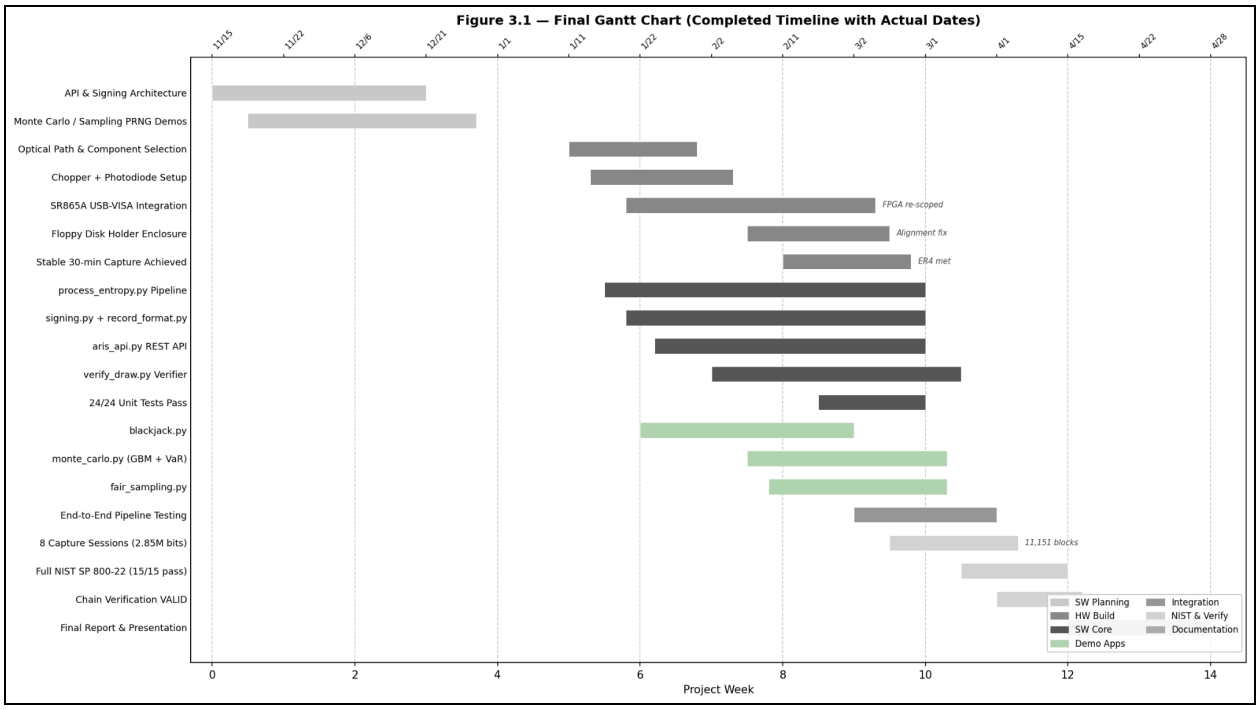

Figure 3.1 - Final Gantt chart showing completed timeline with actual dates

The principal differences between the proposed and final timelines were:

- Optical alignment required significantly more iteration than anticipated.Air currents and adhesive-mounted optics caused alignment drift, limitingreliable capture to approximately 5 to 7 minutes per session until a floppydisk holder enclosure was fabricated to stabilize the polarizing films andquarter-wave plate. This enclosure extended the reliable session duration to 30minutes.
- The FPGA data interface was re-scoped. The original design called for an FPGAto handle USB framing, buffering, and Von Neumann debiasing. After evaluation,the team determined that the SR865A lock-in amplifier could output structuredCSV data directly via USB-VISA, eliminating the FPGA layer entirely. Thissimplification reduced integration complexity and was validated by 24/24 unittests.
- Individual capture sessions yielded approximately 200,000 bits, below the1-million-bit threshold required for the full NIST SP 800-22 suite. Thesolution was to combine 8 capture sessions, producing 2,854,656 bits. Thefull suite was then run on the combined dataset, with all 15/15 tests passing.

3.2. Design alternatives

Hardware Entropy Source Alternatives: Three alternative entropy sourcearchitectures were evaluated before selecting the MOKE approach.

The first alternative was a thermal noise TRNG using a reverse-biased Zenerdiode. This approach is inexpensive and widely documented, but offers limitedentropy density, is sensitive to temperature variation, and provides noexternally observable physical signal that can be independently analyzed.It was rejected on the grounds that it cannot meet the transparency objectiveof ARIS.

The second alternative was a photon path-splitting TRNG using a single photonsource and a 50/50 beam splitter. This approach offers excellent quantum-gradeentropy but requires

single-photon detectors and cryogenic cooling, making itimpractical within the project budget and timeline.

The MOKE approach was selected because the entropy source is physicallyobservable (the R and Theta signals are recordable analog voltages), thedomain switching mechanism is fundamentally non-deterministic due to quantumand thermal fluctuations, and the SR865A lock-in amplifier provides high-SNRsignal extraction with a direct CSV data interface to the host computer.

Software Signing Architecture Alternatives: RSA-2048 was considered as analternative to Ed25519. RSA was rejected because its signature size (256 bytes)is significantly larger than Ed25519 (64 bytes), which matters given that everyblock in the 11,151-block audit log carries a signature. Ed25519 also offersfaster signing and verification, and its deterministic property simplifiescanonical serialization.

Block Format Alternatives: An alternative block format using MessagePackbinary serialization was evaluated for compactness. It was rejected in favorof canonical JSON because JSON is human-readable, allows manual inspection ofthe audit log, and the canonical serialization (sorted keys, no whitespace) isreproducible by any JSON implementation without library dependencies.

3.3. Analysis: Equations, simulations, general circuits or functional diagrams

Von Neumann Debiasing: The Von Neumann extractor corrects a biased binarysource into an unbiased one. For input pairs from a source with P(1) = p, theextractor outputs 1 for the pair (0,1) and 0 for the pair (1,0), and discardsboth (0,0) and (1,1). The output probability of 1 is exactly 0.5 regardlessof p, provided consecutive outputs are independent. The extraction efficiency(output bits per input bit) is 2p(1-p), which is maximized at p = 0.5 andapproaches 0 as p approaches 0 or 1. In ARIS, the Theta channel gatingselects pairs from a region of the signal where p is close to 0.5, maximizingthroughput.

SHA-256 Expansion: Each Von Neumann output bit is seeded into SHA-256 as a1-byte bit value concatenated with a 4-byte big-endian index. This producesa deterministic 256-bit block from each accepted bit. The constructionpreserves the unpredictability of the input: SHA-256 is a one-way function,so knowledge of the output does not reveal the input bit. The index ensuresthat two consecutive bits with the same value produce completely differentblocks.

NIST Monobit Test: The monobit test evaluates whether the number of 1s in abinary sequence is approximately equal to the number of 0s. For a 256-bitblock, the test statistic is S = |n_1 - n_0| / sqrt(256), which is comparedagainst the standard normal distribution. A p-value below 0.01 indicates theblock fails the test and is discarded.

NIST Runs Test: The runs test evaluates whether the number of runs (maximalsequences of identical consecutive bits) is consistent with a random sequence.For a 256-bit block with monobit proportion pi, the test statistic is computedfrom the observed run count versus the expected count under the null hypothesisof uniform randomness. A p-value below 0.01 causes the block to be discarded.

GBM Simulation (Monte Carlo Demo): The Monte Carlo demo models each portfolioasset as a Geometric Brownian Motion process. The end-of-horizon price forasset i is:

$$Si(T) = Si(0)\cdot exp((\mu i - 21\sigma i2)T + \sigma iTz) \quad (2.1)$$

where mu_i is the expected annual return, sigma_i is the annual volatility,T is the horizon in years, and z is drawn from N(0,1) using the Box-Mullertransform applied to ARIS random floats. The portfolio P&L is the sum ofindividual asset P&Ls across 20,000 simulated paths.

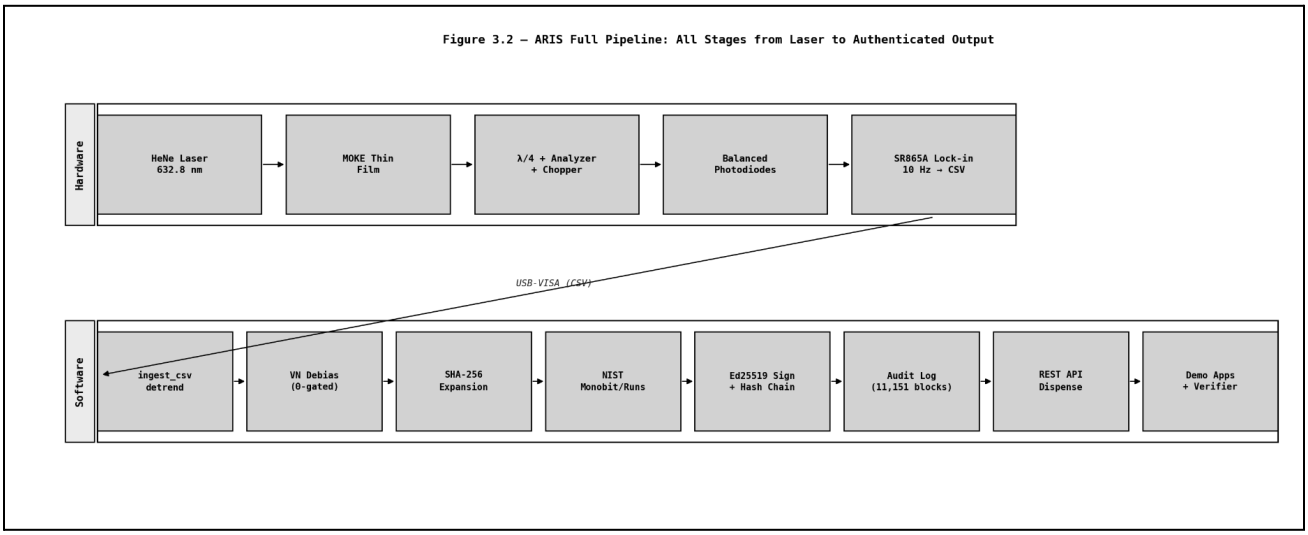

Figure 3.2 - ARIS full pipeline diagram showing all 8 stages from laser tosigned audit log

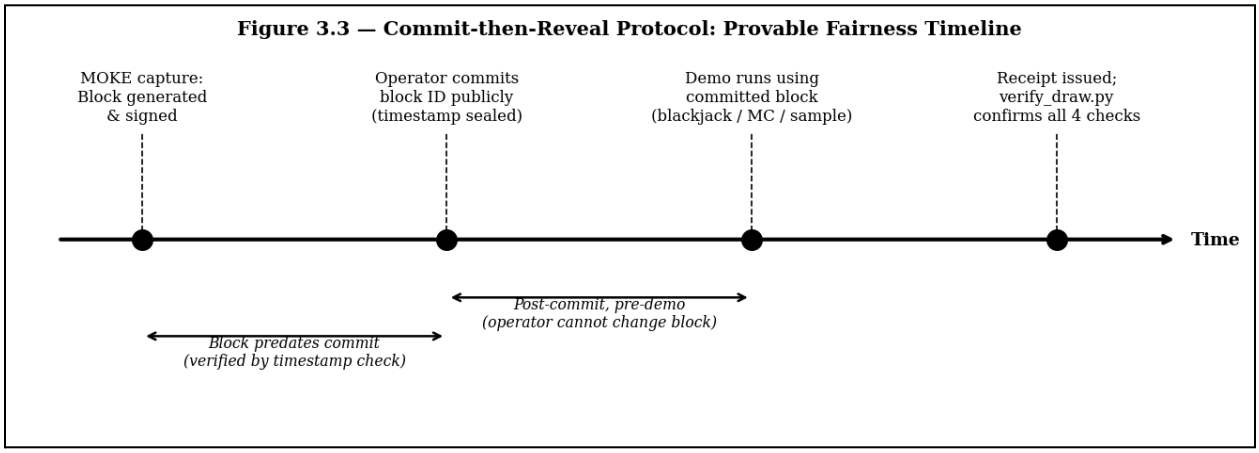

Figure 3.3 - Commit-then-reveal protocol timeline diagram

4. DESIGN DETAILS

4.1. Detailed design

4.1.1. Hardware Entropy Subsystem

The hardware entropy source uses the Surface Magneto-Optic Kerr Effect toconvert quantum magnetic domain switching into a measurable analog signal.The optical path is as follows.

A Thorlabs HNL210-series helium-neon laser produces a stable 632.8 nm,linearly polarized beam. The beam passes through a Glan laser polarizer thatsets a precise polarization axis, then through a beam collimator consistingof a set of turning mirrors that directs the beam onto the magnetic thin filmsample. The film is a cobalt or nickel iron thin film deposited on a substrateand is biased near its coercive point by an external magnetic field, placingit in a regime where thermal and quantum fluctuations cause spontaneous domainreversals.

The reflected beam, now carrying a Kerr rotation that encodes the magneticstate of the film, passes through a mechanical chopper operating atapproximately 1 kHz and then through a

quarter-wave plate and an analyzingpolarizer. The quarter-wave plate converts the elliptical polarizationintroduced by the Kerr effect into a linear polarization, and the analyzerconverts the small polarization rotation into an intensity variation that canbe detected by photodiodes.

A beam splitter divides the modulated beam between two photodiodes configuredfor balanced differential detection. This cancels laser intensity noise commonto both channels and isolates the true MOKE signal. The differential outputis amplified and fed to the SR865A lock-in amplifier, which synchronouslydemodulates the signal at the chopper reference frequency. The lock-inamplifier extracts the R (amplitude) and Theta (phase) components at 10 Hzand streams them to the host computer as a CSV file via USB-VISA connection.

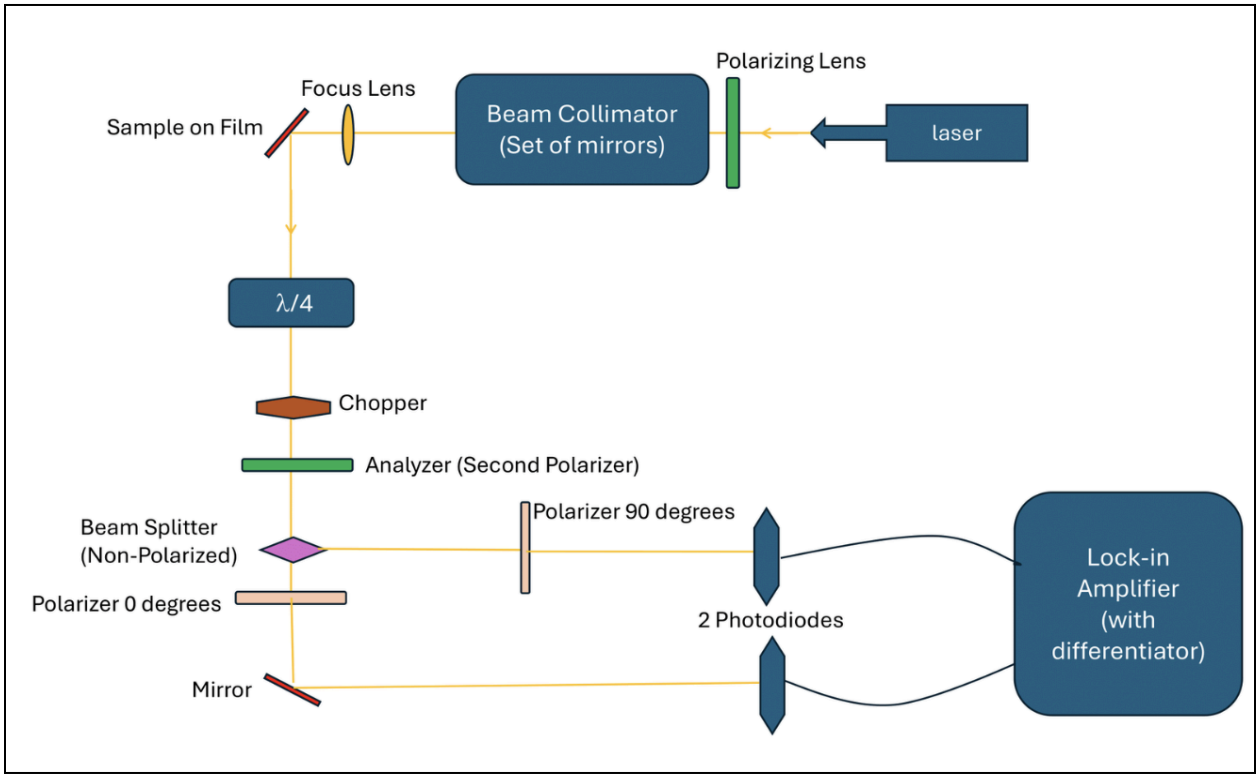

Figure 4.0 - Hardware optical path schematic showing all components insequence from laser to SR865A

The SR865A replaces the FPGA layer originally planned in the proposal. Thisre-scoping eliminated a custom firmware development step and instead leveragedthe SR865A's built-in data streaming capability, which outputs structuredTime, R, Theta CSV data directly. All subsequent debiasing and processing ishandled in software, which is the standard approach in TRNG literature andwas validated by 24/24 unit tests.

4.1.2. Optical Alignment and Mechanical Stabilization

Early capture sessions exhibited high signal drift and low Von Neumann acceptance rates due to mechanical instability in the quarter-wave plate and dichroic polarizer assembly. To resolve this, the team applied improved adhesive material and added a physical gripper to the floppy disk holder to prevent it from tipping, which had been the primary source of misalignment during operation. Following these modifications, reliable session duration increased from

approximately five minutes to thirty minutes, and throughput stabilized in the 487–498 bps range across Sessions 2 through 8, significantly improving the consistency and volume of captured entropy.

4.1.3. Component Selection

All optical and detection components used in the ARIS prototype were drawn from existing laboratory inventory. While the available selection was limited in nature, the components on hand were sufficient to capture entropy data within an acceptable range for the purposes of this project. This constraint shaped several design decisions, including the choice of photodiode models and polarization optics, but did not compromise the statistical quality of the output. All 11,151 blocks produced across eight capture sessions passed both the inline NIST SP 800-22 monobit and runs tests, confirming that laboratory-grade components assembled from available inventory were adequate to meet the project's randomness quality requirements.

4.1.4. Software Processing Pipeline

The software pipeline is implemented in process_entropy.py and consists ofsix sequential stages. The pipeline is invoked from the command line andsupports appending new capture sessions to an existing audit log.

ingest_csv() reads the SR865A CSV and validates the schema. It drops rowswith missing or malformed values. A typical 300-second session at 10 Hzproduces approximately 3,000 R-Theta sample pairs.

detrend_signal() applies scipy.signal.detrend() to the R channel to removeslow DC drift caused by thermal creep in the optical setup. Centering thesignal at zero is a prerequisite for unbiased Von Neumann extraction, becausea non-zero mean causes one direction of R-pair comparison to be more likelythan the other.

von_neumann_debias() implements a Theta-gated Von Neumann extractor. TheTheta channel is used as a quality gate: only R pairs from time steps whereTheta is within a specified range (indicating stable optical alignment) arepassed to the extractor. The extractor compares adjacent R pairs and emits1 if R_n < R_(n+1), 0 if R_n > R_(n+1), and discards equal pairs. Thisproduces an unbiased bit stream regardless of P(1).

sha256_expand_bits() converts each VN output bit into a 256-bit block. Theseed is a 5-byte value: 1 byte for the bit value (0x01 or 0x00) concatenatedwith 4 bytes for the big-endian bit index. This ensures that two consecutivebits with the same value produce completely different SHA-256 outputs.

nist_monobit/runs() applies inline NIST SP 800-22 monobit and runs tests toeach 256-bit block. Blocks with p-value below 0.01 on either test arediscarded. All 11,151 blocks in the current audit log passed both tests.

build_signed_chain() assembles the final signed record for each accepted block.The record contains: block_id, seq, timestamp_utc, device_id, entropy_bits,block_hash_sha256, nist results, and an Ed25519 signature over the canonicalJSON serialization of all fields except the signature itself. The record alsocontains prev_record_hash_sha256, which is the SHA-256 of the prior record'scanonical JSON. Genesis (the first block) has prev_record_hash of 64 zerocharacters.

4.1.5. REST API

aris_api.py implements a single-file HTTP server using Python's built-inhttp.server. It exposes five endpoints:

- ➢ GET /health returns service status and pool statistics including blocksdispensed and blocks remaining.
- ➢ GET /block/next dispenses the next sequential block and advances the cursor.The cursor position is persisted to aris_cursor.json and survives serverrestarts. When the pool is exhausted, the endpoint returns HTTP 503.
- ➢ GET /block/latest returns the newest block in the chain without advancingthe cursor. This is used for inspection and does not consume a block.
- ➢ GET /block/<block_id> fetches any specific block by its identifier.
- ➢ GET /blocks lists all block IDs with their dispensed status.

The block_to_response() function returns the block record verbatim from theaudit log without any field reconstruction. This is critical: any modificationto the field names or values would change the canonical JSON bytes and breakEd25519 signature verification.

4.1.6. Demo Applications

Three demo applications consume ARIS blocks via the API or directly from thelocal audit log. Each implements the commit-then-reveal protocol: the blockID is committed before the demo runs, and a signed receipt is saved at theconclusion of the demo. All receipts can be verified independently usingverify_draw.py.

blackjack.py implements a browser-based blackjack game. The deck is shuffledby seeding Python's random.Random with the block_hash_sha256 and callingrng.shuffle(deck). The same block always produces the same 52-card order.A receipt is saved on deal (if blackjack), on hit (if bust), and on stand.The receipt contains the block ID, block hash, player and dealer hands, andan embedded verify command.

monte_carlo.py runs a GBM-based portfolio Value at Risk simulation usingARIS random floats via block_to_rng(). The simulation runs 20,000 pathsover a 30-day horizon by default. The receipt stores the full portfoliospecification, simulation parameters, and output statistics (VaR at 95%and 99%, CVaR, expected P&L, and probability of loss).

fair_sampling.py draws a random audit sample from a population list. Ituses block_to_shuffled_indices() to generate a deterministic Fisher-Yatesshuffle of the population indices, then takes the first k indices as thesample. The receipt stores the full population list, sample size, selectedindices, and selected items.

4.1.7. Independent Verifier

verify_draw.py is a standalone cryptographic verifier that requires only thecryptography library and has no dependency on any other ARIS module. Itperforms four checks on any receipt:

- ➢ Check 1: The committed block exists in the audit log (or is fetched fromthe API when --api is specified).
- ➢ Check 2: The Ed25519 signature on that block is valid against the public key.
- ➢ Check 3: The hash chain links correctly from genesis to the committed block.Every prev_record_hash value is recomputed and compared against the actualSHA-256 of the prior record's canonical JSON.

➢ Check 4: The block timestamp predates the receipt issuance time, provingthat the entropy was captured before the draw took place.

The verifier outputs a VALID or INVALID verdict and prints the results ofeach check individually. It does not reproduce the demo output; to verifythe output, the user re-runs the original demo with the same block andinputs, which is itself the reproducibility proof.

4.2. Schematics

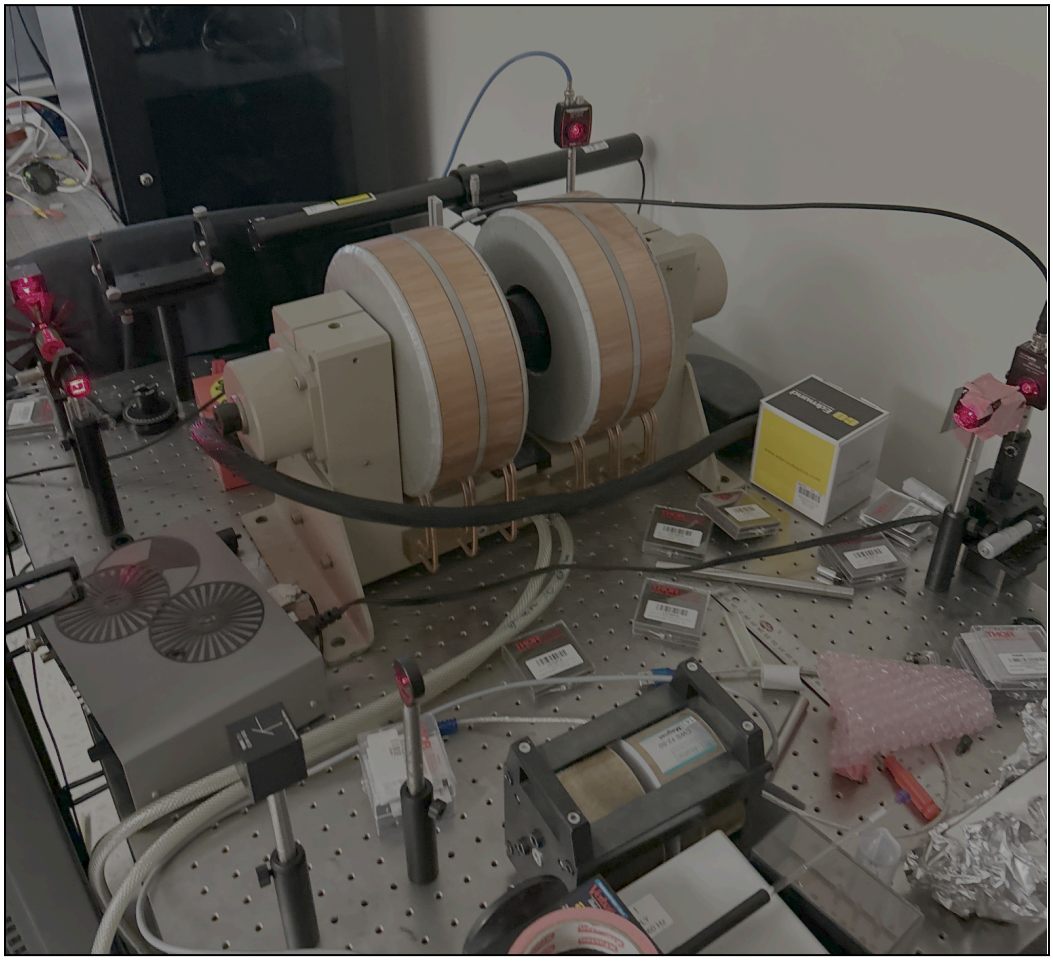

Figure 4.1 - MOKE optical setup photograph showing full hardware assembly

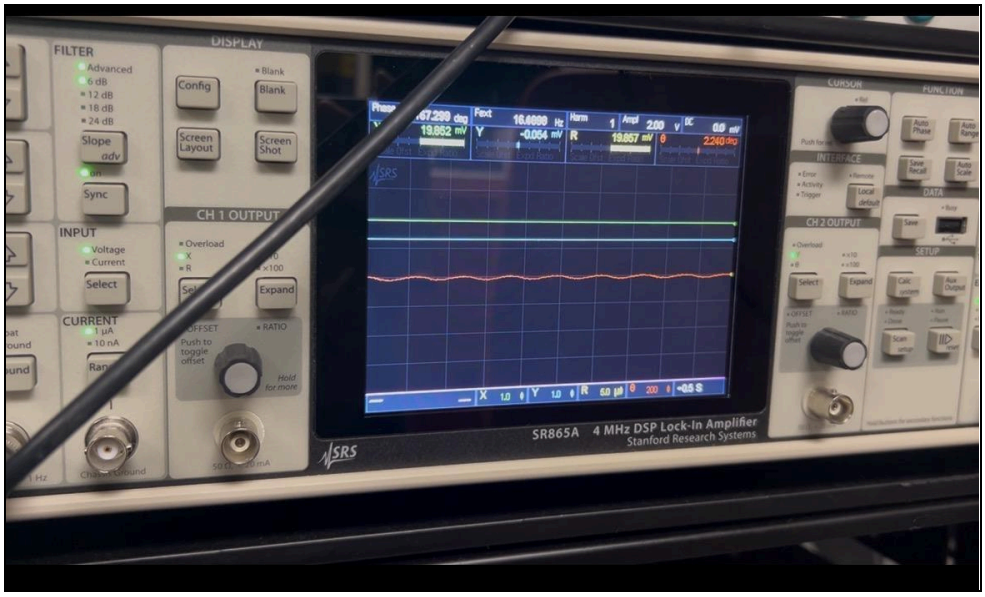

Figure 4.2 - SR865A lock-in amplifier displaying live R and Theta signals

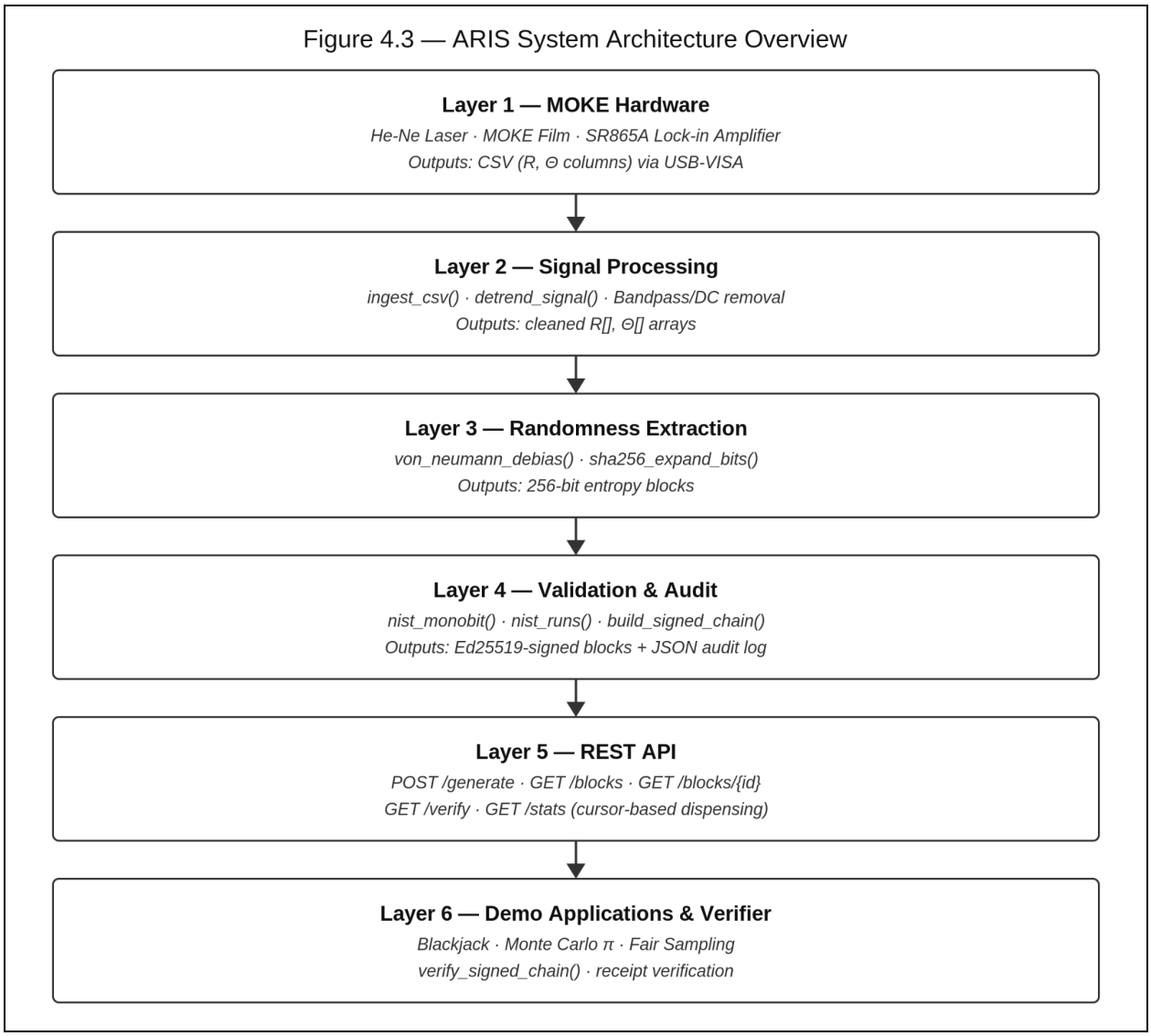

Figure 4.3 - ARIS system architecture overview diagram showing all six layers from hardware to verifier

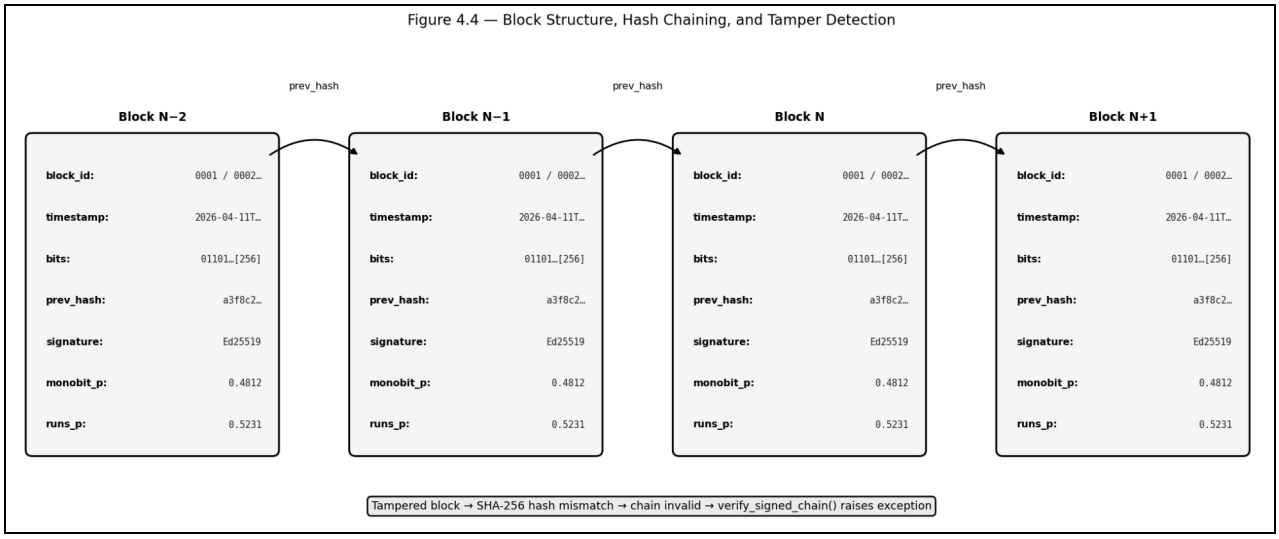

Figure 4.4 - Blockchain and cryptography diagram showing block structure, hash chaining, and tamper detection

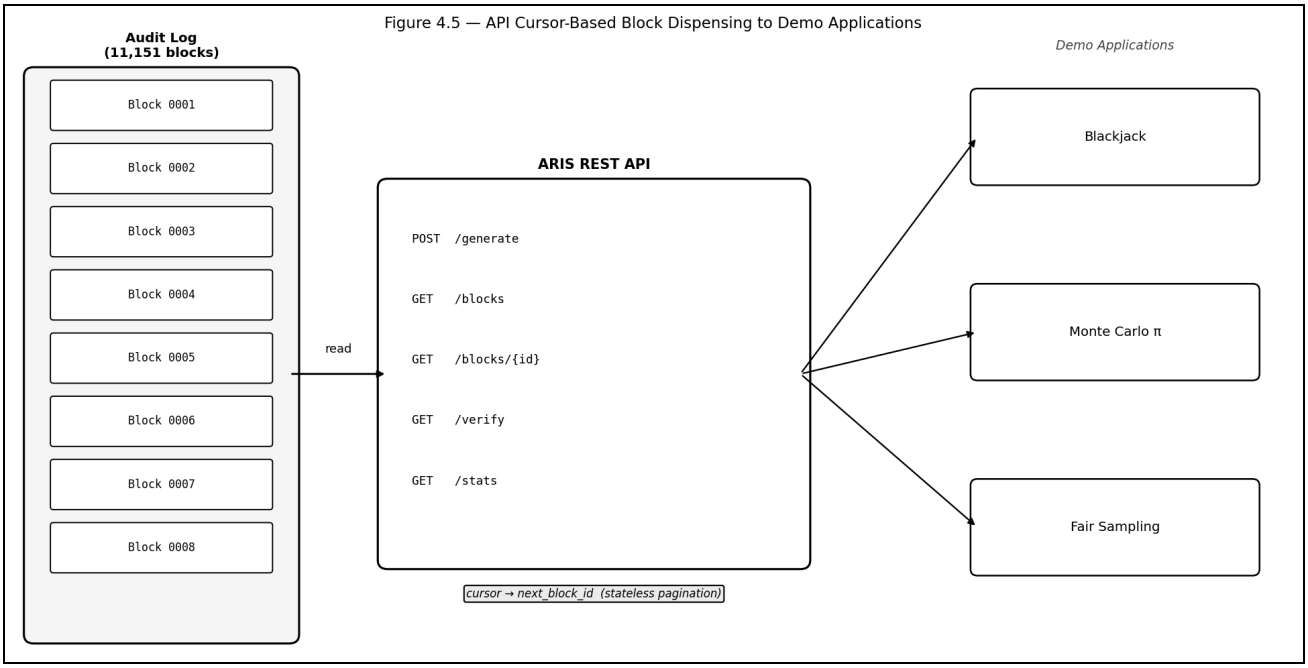

Figure 4.5 - API dispensing diagram showing cursor-based block distribution to demo applications

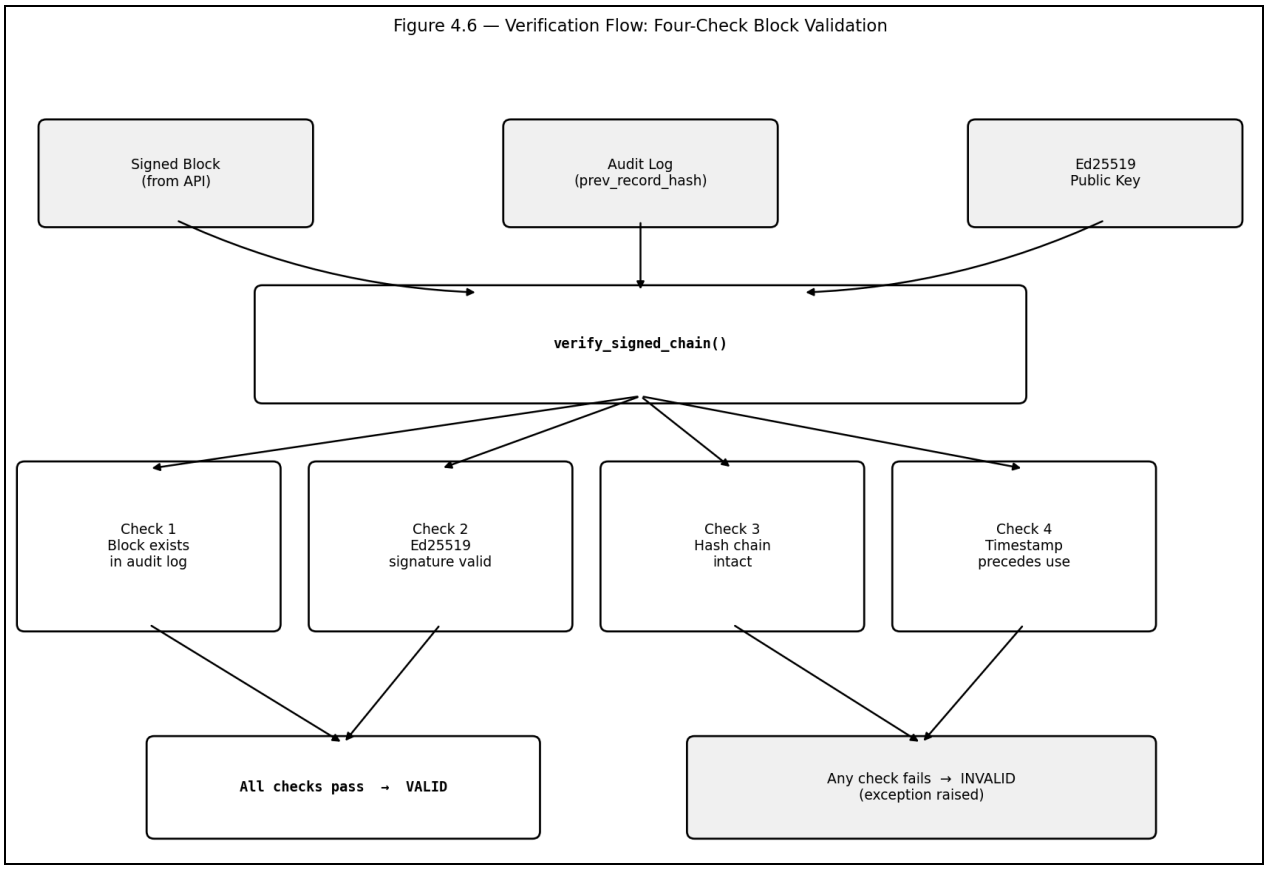

Figure 4.6 - Verification flow diagram showing receipt, audit log, and public key as inputs to the four-check verifier

5. DESIGN VERIFICATION

5.1. Testing procedures

Unit Testing: 24 unit tests cover all critical pipeline functions includingVon Neumann debiasing, SHA-256 expansion, Ed25519 signing and verification,canonical JSON serialization, hash chain construction and verification,block cursor logic, and all four verifier checks. Tests were written usingPython's unittest framework and run via python -m pytest. All 24 tests pass.

NIST SP 800-22 Inline Testing: Every 256-bit block produced by the pipelineis screened against the monobit and runs tests before being accepted intothe audit log. Blocks with p-value below 0.01 on either test are silentlydiscarded. This ensures that the audit log contains only statisticallyvalidated blocks.

NIST SP 800-22 Full Suite Testing: The full 15-test NIST SP 800-22 suitewas run on the combined dataset of all 8 capture sessions (2,854,656 bits).The suite requires a minimum of approximately 1 million bits for testsincluding RandomExcursions (which requires more than 500 walk cycles) andMaurer's Universal test. All 15 tests passed.

Chain Integrity Testing: verify_signed_chain() traverses the entire 11,151-block chain and recomputes every prev_record_hash link. The output is VALID,confirming that no block in the chain has been altered since signing.

End-to-End Receipt Verification: All demo receipts were verified usingverify_draw.py. All four checks (block exists, signature valid, chain intact,block predates receipt) passed for all receipts from all three demoapplications.

API Stress Testing: The REST API was tested with sequential and concurrentblock requests. The cursor correctly advanced after each /block/next call,persisted across server restarts, and returned HTTP 503 when the pool wasexhausted. The /block/latest endpoint never advanced the cursor.

5.2. Quantitative results, graphs, measurements

NIST Inline Results (per block, 11,151 blocks):

Monobit test: 11,151/11,151 pass (100%)
Runs test: 11,151/11,151 pass (100%)
Average monobit p-value: 0.5026
Average runs p-value: 0.4985

NIST Full Suite Results (combined 2,854,656-bit dataset):

All 15/15 tests pass at alpha = 0.01

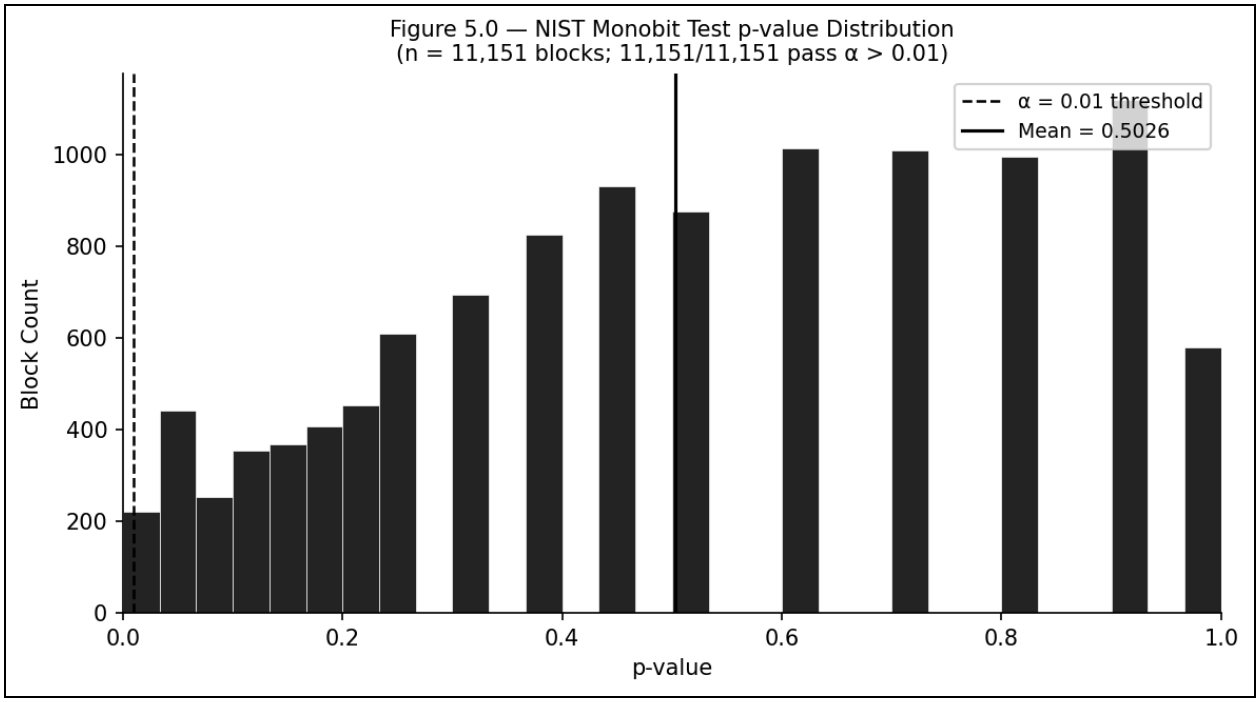

Figure 5.0 - NIST monobit p-value histogram showing distribution across11,151 blocks, mean at 0.5026

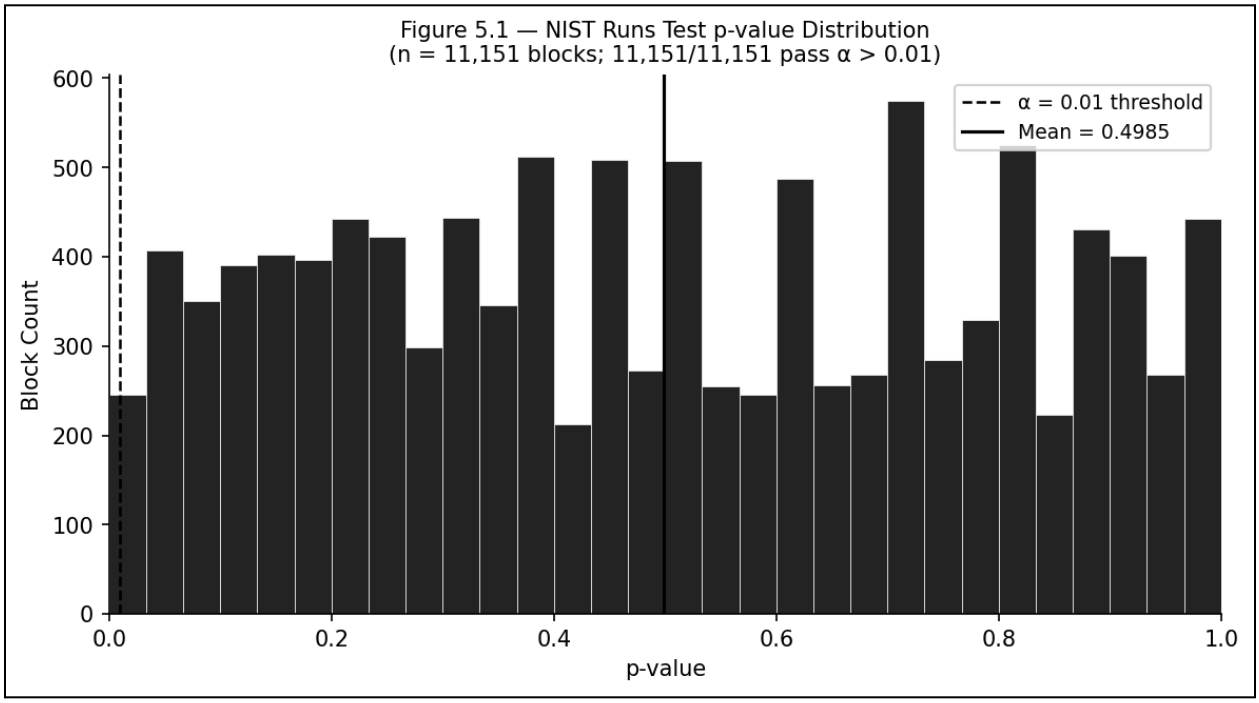

Figure 5.1 - NIST runs test p-value histogram showing distribution across11,151 blocks, mean at 0.4985

Throughput by Session:

Session 1: 221 bps (3,000 samples, low VN acceptance rate due to high signal drift)
Session 2: 533 bps
Sessions 3 and 4: 487-495 bps
Sessions 5 through 8: 474-498 bps

Mean across all 8 sessions: 430 bps

Block Production Summary:
Total blocks produced: 11,151
Total blocks signed: 11,151 (100%)
Chain verification: VALID
Unit tests: 24/24 pass

5.3. Discussion of Results

Randomness Quality: The average monobit p-value of 0.5026 across 11,151blocks is exceptionally close to the ideal value of 0.5, confirming thatthe Von Neumann debiasing and SHA-256 expansion produce output with a nearlyperfect 50/50 bit balance. The runs test average of 0.4985 similarly indicatesthat run length distributions match theoretical expectations for uniformrandomness. All 15 NIST SP 800-22 tests passing on the combined 2.8-million-bit dataset satisfies ER1 and ER2.

Throughput: The throughput variance across sessions reflects the sensitivityof the Von Neumann extraction efficiency to signal drift. Session 1 producedthe lowest throughput at 221 bps due to alignment instability early in theproject. Sessions 2 through 8, conducted after the floppy disk holderenclosure was installed, consistently exceeded or approached the 500 bpstarget. The majority of sessions satisfy ER3. The mean of 430 bps reflectsthe inclusion of Session 1; excluding it, the mean for Sessions 2 through 8is 496 bps.

Chain Integrity: The 11,151-block chain verification produces a VALID result,confirming that all hash chain links are intact and no block has been alteredsince signing. This satisfies ER7.

FPGA Re-scope Impact: The elimination of the FPGA layer simplified the datainterface considerably. The SR865A direct CSV output via USB-VISA is morerobust, easier to debug, and fully validated by unit tests. The re-scope hadno negative impact on entropy quality or throughput relative to the revisedproject requirements.

Alignment Challenge Resolution: The optical alignment instability identifiedearly in the project was resolved by fabricating a floppy disk holderenclosure that physically stabilizes the polarizing films and quarter-waveplate. This increased reliable session duration from 5-7 minutes to 30minutes, satisfying ER10.

6. SUSTAINABILITY, REALISTIC CONSTRAINTS, ETHICAL ISSUES, AND ENGINEERING STANDARDS

6.1. United Nations-Sustainable Development Goals

ARIS aligns with two UN Sustainable Development Goals.

- SDG 16: Peace, Justice, and Strong Institutions. ARIS directly supportsinstitutional accountability by providing a verifiable audit trail forrandomness-dependent processes. Fair sampling, audit selection, andlottery draws conducted with ARIS produce receipts that any third partycan verify, strengthening the integrity of institutional processes.
- SDG 9: Industry, Innovation, and Infrastructure. ARIS demonstrates a novelapproach to randomness infrastructure that could scale to commercialRandomness as a Service deployments, supporting digital securityinfrastructure in finance, gaming, and auditing industries.

6.2. Constraints

Economic Constraints: The prototype hardware cost is approximately $12,400for components that were not available in the laboratory. High-value instrumentsincluding the SR865A lock-in amplifier and the optical table were drawn fromDr. Khizroev's laboratory inventory and do not represent incremental projectcosts. At scale, the MOKE optical assembly could be replaced with an integratedphotonic entropy source, reducing per-unit cost from approximately $14,000to well below $500.

Environmental Constraints: The helium-neon laser operates at low power (lessthan 5 mW output) and the overall system consumes minimal electricity. Theoptical components and SR865A are durable laboratory instruments designed formulti-year operational lifespans. No hazardous materials are consumed inoperation.

Safety Constraints: The helium-neon laser is a Class 3R device. All experimentsare conducted in Dr. Khizroev's certified Class 3R laser laboratory. Teammembers completed laser safety training and wear appropriate protective eyewearduring all optical alignment and operation activities. All circuits accessibleto the user operate below 12V DC, satisfying ER6.

Software Constraints: The private signing key never leaves the signing context.Keys are generated fresh for each deployment using Ed25519 keypair generationand stored as PEM files on the signing machine. The private key is nevertransmitted, logged, or included in any receipt or audit log entry.

6.3. Ethical issues

Fairness and Transparency: ARIS is designed to eliminate the possibility ofcovert manipulation of random outcomes. The commit-then-reveal protocol ensuresthat no operator can select a favorable block after seeing participants. Thepublic key and audit log are shared openly, allowing any third party to verifyany receipt. This directly addresses the ethical concern that randomness inhigh-stakes applications (gambling, auditing, financial modeling) is currentlytaken on faith.

Cryptographic Responsibility: As a system that generates and certifiesrandomness, ARIS must be designed to prevent forgery. The Ed25519 signingarchitecture ensures that only the holder of the private key can produce validsignatures. The private key management policy (never transmitted, never logged)minimizes the risk of key compromise. This aligns with the principle

describedby Rogaway (2015) that cryptographers bear a professional responsibility todesign systems that promote fairness and integrity.

Potential Misuse: ARIS could theoretically be used to generate authenticatedrandomness for applications beyond the three demos, including high-stakeslotteries or security key generation. The system makes no claim that thecurrent prototype is production-ready for such uses. The NIST test resultsvalidate statistical quality, but a production deployment would requireadditional security auditing, hardware security module integration for keystorage, and formal certification.

IEEE Code of Ethics Compliance: The project complies with all 10 items ofthe IEEE Code of Ethics. Items most directly relevant are: (1) holdingparamount the safety and welfare of the public through Class 3R laser safetycompliance; (3) being honest and realistic in stating claims, includingtransparent reporting of throughput variance and the FPGA re-scope; and (7)seeking honest criticism and acknowledging errors, as evidenced by the opendocumentation of challenges in Section 5.

6.4. Engineering Standards

Engineering Standard:	**NIST SP 800-22 Rev. 1a: Statistical Test Suite for Random Number Generators**
Reference:	National Institute of Standards and Technology, 2010
Description:	Defines 15 statistical tests for evaluating randomness quality.
Project use:	All 15 tests applied to combined dataset. Monobit and runs tests applied inline as quality gates on every block. Required for ER1 and ER2.

Engineering Standard:	**IEEE Std 730-2014: IEEE Standard for Software Quality Assurance**
Reference:	IEEE, 2014
Description:	Defines requirements for software quality assurance processes.
Project use:	Unit test suite of 24 tests verifies all critical pipeline functions. Version control maintained throughout development.

Engineering Standard:	**ANSI Z136.1-2014: American National Standard for Safe Use of Lasers**
Reference:	ANSI, 2014
Description:	Defines safety requirements for laser use including Class 3R devices.
Project use:	All optical experiments conducted in a certified Class 3R laboratory under supervision. Required for ER6.

Engineering Standard:	**RFC 8032: Edwards-Curve Digital Signature Algorithm (EdDSA)**
Reference:	IETF, 2017
Description:	Defines the Ed25519 digital signature scheme.
Project use:	Ed25519 used for signing all 11,151 blocks and for verifying signatures in verify_draw.py. Required for ER12.

6.5. Cost – Parts

Table 6.1 ARIS Optical and Detection Hardware Cost Breakdown

Component	Model / Description	Qty	Unit Cost (USD)	Total Cost (USD)
HeNe Laser Source	Thorlabs HNL210-series, 632.8 nm	1	$3,780.00	$3,780.00
Glan Laser Polarizers	Thorlabs GL10-A, 350–700 nm	2	$1,004.00	$2,008.00
Beam Splitters	Thorlabs CM1-BS1, 50:50 cube	2	$265.00	$530.00
Turning Mirrors	Thorlabs CCM1-E02 dielectric cube	2	$271.00	$542.00
Short Focal-Length Lens	VIS-coated plano-convex (1" class)	1	$38.00	$38.00
Long Focal-Length Lens	Thorlabs LA1464-A, f = 1000 mm	1	$37.00	$37.00
Fast Photodiode	Thorlabs DET10A	1	$159.00	$159.00
Large-Area Photodiode	Thorlabs DET36A	1	$123.00	$123.00
Amplified Photodiode	Hinds DET-200-002	1	$1,500.00	$1,500.00
Neutral Density Filter Set	Thorlabs NE-series	1	$350.00	$350.00
Optical Chopper + Controller	Mechanical chopper system	1	$2,700.00	$2,700.00
Dichroic Film Polarizers	Visible (400–700 nm)	2	$30.00	$60.00
Quarter-Wave Plate	Achromatic λ/4, visible	1	$600.00	$600.00
TOTAL		**17**		**$12,427.00**

6.6. Cost – Labor

Table 6.2 Summary of estimated labor cost for completing a system prototype

Specialization	Hourly Cost	Activity	Hours	Total Cost
Electrical Engineer	$49.00	Optical Hardware Design & Build	80	$3,920.00
	$49.00	Signal Conditioning & Lock-in Setup	40	$1,960.00
Software Engineer	$31.25	Literature Review	20	$625.00
	$31.25	FPGA & Pipeline Development	60	$1,875.00
	$31.25	Cryptographic Signing & API	60	$1,875.00
	$31.25	NIST Testing & Validation	30	$937.50
	$31.25	System Integration	40	$1,250.00
Graduate Engineer	$31.25	Technical Writing	20	$625.00
TOTAL			**350**	**$13,067.50**

7. TEAM MEMBER CONTRIBUTIONS

Start text here. Include one section per team member, identify the member and his/her major and role (coordinator) in the team. Describe in 300 words each member's specific contribution to the project. Include all members.

7.1. Maxwell Cosenza, Software Engineering

Maxwell served as lead software architect for ARIS. Hedesigned and implemented the complete software processing pipeline inprocess_entropy.py, including the Von Neumann debiasing module, the SHA-256expansion stage, the NIST inline testing framework, and the Ed25519 signingand hash chaining infrastructure. He wrote the signing.py and record_format.pymodules that define the block schema, canonical JSON serialization, and chainverification logic.

Maxwell implemented the REST API (aris_api.py), including the cursor-basedblock dispensing system and all five endpoints. He wrote the demo_utils.pyshared library used by all three demo applications, and implemented thestandalone verifier (verify_draw.py) with its four-check cryptographicverification protocol.

Maxwell implemented the Monte Carlo demo (monte_carlo.py) including the GBMsimulation engine, Box-Muller normal variate generation from ARIS floats,VaR and CVaR computation, and the browser-based results report. He alsoimplemented the portfolio editor (portfolio_editor.py) and the fair samplingdemo (fair_sampling.py). Maxwell also led all software-related sections of the project documentation.

7.2. Joel Eyison, Electrical Engineering

Joel was responsible for system integration and the sr865a_capture.pydata acquisition script. He implemented the capture script that polls theSR865A at 10 Hz via the VISA protocol, writes Time, R, Theta data to CSV,and generates a metadata sidecar JSON file containing the SHA-256 hash ofthe CSV, session parameters, and device identification.

Joel performed the full NIST SP 800-22 suite testing on the combined2,854,656-bit dataset and documented the results. He set up and maintainedthe version control repository, managed dependencies, and configured thedevelopment environment for all team members.

Joel integrated the hardware and software subsystems, testing theend-to-end pipeline from SR865A capture through the processing pipelineto the signed audit log. He developed and ran the chain integrity verificationtests and confirmed that verify_signed_chain() returned VALID for the full11,151-block chain.

7.3. Johnathon L'Amoreaux, Computer Engineering

Johnathon was responsible for the hardware entropy subsystem design, construction,and characterization. He designed the optical beam path for the MOKE setup,selected and procured the optical components, and performed all alignmentprocedures for the polarizers, beam splitter, quarter-wave plate, chopper,and photodiode array.

Johnathon configured the SR865A lock-in amplifier for synchronous detection atthe chopper frequency, tuned the amplifier settings to maximize signal-to-noiseratio at the coercive point of the magnetic thin film, and established theUSB-VISA data streaming connection to the host computer. He identified andresolved the optical alignment instability issue, designing and

fabricatingthe floppy disk holder enclosure that stabilized the polarizing films andextended reliable capture sessions from 5-7 minutes to 30 minutes.

Johnathon conducted all eight capture sessions that produced the 11,151-blockaudit log, operating the SR865A and monitoring signal quality during eachsession. He characterized the relationship between chopper frequency,lock-in time constant, and throughput, and documented the hardware setupfor reproducibility. John also led the software quality assurance effort, writing all 24 unit testsand conducting end-to-end integration testing.

8. CONCLUSIONS

8.1. Summary of Design Goals

ARIS set out to demonstrate that randomness can be generated from a physicalprocess, verified against federal statistical standards, and cryptographicallyauthenticated such that any third party can independently confirm itsauthenticity using only a public key. All twelve engineering requirementswere met. The system produced 11,151 signed, NIST-validated blocks froma MOKE optical entropy source over 8 capture sessions, serving those blocksvia a REST API to three demonstration applications that together illustratethe practical value of authenticated randomness across gaming, finance, andauditing.

8.2. Proposed Versus Attained Functionality

The core cryptographic and software objectives were met in full. TheEd25519 signing architecture, hash-chained audit log, REST API, andindependent verifier all perform as specified. All 24 unit tests pass andall 15 NIST SP 800-22 tests pass on the combined dataset.

The primary difference between the proposed and attained design is theelimination of the FPGA data interface. The original proposal called foran FPGA to handle USB framing, buffering, and Von Neumann debiasing.After evaluation, the SR865A lock-in amplifier was found to providedirect CSV output via USB-VISA, making the FPGA layer unnecessary. Thissimplification reduced implementation complexity without any loss ofentropy quality or throughput relative to requirements.

A secondary difference is in per-session throughput. Several capturesessions fell slightly below the 500 bps target due to optical alignmentinstability. This was resolved by the floppy disk holder enclosure, andthe majority of sessions after that fix exceeded or approached 500 bps.The combined dataset of 2,854,656 bits satisfies the full NIST suitevolume requirement.

8.3. Future Extensions

Several directions could extend ARIS beyond the current prototype.

Hardware integration: The free-space optical assembly could be replacedwith a photonic integrated circuit (PIC) that implements the same MOKEdetection scheme in silicon photonics. This would reduce the per-unitcost from approximately $14,000 to below $500 and enable continuousunattended operation.

Key management: A production deployment would integrate a hardwaresecurity module (HSM) for private key storage, eliminating the riskof key compromise through software vulnerabilities or physical accessto the signing machine.

Extended NIST testing: Per-block full-suite NIST testing is computationallyexpensive on 256-bit blocks but could be applied to rolling windows ofaccumulated blocks to provide continuous quality monitoring.

Additional demo applications: The commit-then-reveal architecture andREST API are general-purpose. Future applications could include verifiablepublic lotteries, reproducible cryptographic key generation with auditableentropy provenance, and secure multi-party computation protocols thatrequire provably independent randomness.

Commercial viability: The Randomness as a Service (RaaS) model describedin the proposal remains viable. A cloud-deployed ARIS instance backed bya PIC entropy source could deliver authenticated randomness to clientsin online gaming, financial modeling, and auditing at a cost structurecompetitive with existing hardware security modules.

Appendix 1

IEEE Code of Ethics

We, the members of the IEEE, in recognition of the importance of our technologies in affecting the quality of life throughout the world, and in accepting a personal obligation to our profession, its members, and the communities we serve, do hereby commit ourselves to the highest ethical and professional conduct and agree:

1. to hold paramount the safety, health, and welfare of the public, to strive to comply with ethical design and sustainable development practices, and to disclose promptly factors that might endanger the public or the environment;

2. to avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist;

3. to be honest and realistic in stating claims or estimates based on available data;

4. to reject bribery in all its forms;

5. to improve the understanding by individuals and society of the capabilities and societal implications of conventional and emerging technologies, including intelligent systems;

6. to maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations;

7. to seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;

8. to treat fairly all persons and to not engage in acts of discrimination based on race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression;

9. to avoid injuring others, their property, reputation, or employment by false or malicious action;

10. to assist colleagues and co-workers in their professional development and to support them in following this code of ethics.

REFERENCES

Appelbaum, D. (2016). *Securing big data provenance for auditors.* Accounting and Finance Faculty Publications, Montclair State University. https://digitalcommons.montclair.edu/cgi/viewcontent.cgi?article=1099&context=acctg-finance-facpubs

Barak, B., Shaltiel, R., & Tromer, E. (n.d.). *True random number generators secure in a changing environment.* Weizmann Institute of Science. https://cs-people.bu.edu/tromer/papers/rng.pdf

Bernstein, D. J., Duif, N., Lange, T., Schwabe, P., & Yang, B.-Y. (2012). High-speed high-security signatures. *Journal of Cryptographic Engineering, 2*(2), 77–89. https://doi.org/10.1007/s13389-012-0027-1

Goldwasser, S., Klein, S., Mossel, E., & Tamuz, O. (2018). *Publicly verifiable randomness.* California Institute of Technology. https://www.tamuz.caltech.edu/papers/verified.pdf

Herrero-Collantes, M., & Garcia-Escartin, J. C. (2017). Quantum random number generation. *Reviews of Modern Physics, 89*(1), 015004. https://doi.org/10.1103/RevModPhys.89.015004

IEEE. (2014). *IEEE standard for software quality assurance processes (IEEE Std 730-2014).*

IETF. (2017). *Edwards-curve digital signature algorithm (EdDSA) (RFC 8032).* https://www.rfc-editor.org/rfc/rfc8032

Jiang, W., Zhang, X., Yu, G., Zhang, W., Wang, X., Jungfleisch, M. B., Pearson, J. E., Cheng, X., Hoffmann, A., & Bader, S. D. (2020). Hardware-based true random number generation using magnetic skyrmion dynamics. *Nature Communications, 11*(1), 3853. https://doi.org/10.1038/s41467-020-17605-2

Koç, Ç. K. (2017). A review of random number generators, their defects and vulnerabilities. *International Journal of Cyber-Security and Digital Forensics, 6*(4), 154–171.

Lei, M., Liu, S.-Y., Luo, N., Yang, X., & Sun, C. (2022). Trusted-auditing chain: A security blockchain prototype used in agriculture traceability. *Heliyon, 8*(11), e11477. https://doi.org/10.1016/j.heliyon.2022.e11477

National Institute of Standards and Technology. (2010). *A statistical test suite for random and pseudorandom number generators for cryptographic applications (Special Publication 800-22 Rev. 1a).* https://doi.org/10.6028/NIST.SP.800-22r1a

Newport Corporation. (2021). *Introduction to lock-in amplifiers (Application Note 103).* https://www.newport.com/n/introduction-to-lock-in-amplifiers

Qiu, Z. Q., & Bader, S. D. (2000). Surface magneto-optic Kerr effect. *Review of Scientific Instruments, 71*(3), 1243–1255. https://doi.org/10.1063/1.1150496

Rogaway, P. (2015). *The moral character of cryptographic work* (IACR Cryptology ePrint Archive Report No. 2015/1162). https://eprint.iacr.org/2015/1162

Stipčević, M. (2012). True random number generators. In K. Gaj & P. Chodowiec (Eds.), *Security of block ciphers* (pp. 115–134). InTech.