FMDB *Transactions on Sustainable Finance and Data Science*

Cryptographic Policy Twins for End-to-End Data Lifecycle Integrity in Regulated Information Systems

R. Senthilkumar[1,*], S. Kamalakannan[2], K. Balakrishnan[3], J. Senthil[4], Asique Kunhalakath[5]

[1]Department of Artificial Intelligence and Data Science, Shree Venkateshwara Hi-Tech Engineering College, Erode, Tamil Nadu, India.
[2,3]Department of Electronics and Communication Engineering, Shree Venkateswara Hi-Tech Engineering College, Erode, Tamil Nadu, India.
[4]Department of Artificial Intelligence and Data Science, Centre for Quantum Computing, Kalaignarkarunanidhi Institute of Technology, Coimbatore, Tamil Nadu, India.
[5]Department of Information Technology and Digital Transformation, The Zubair Corporation LLC, Muscat, Muscat Governorate, Oman.
svhecrsk@gmail.com[1], rskama09@gmail.com[2], balakrishnanvp@gmail.com[3], senthilgoi05@gmail.com[4], asque.farhanth@gmail.com[5]

Abstract: In this paper, researchers introduce the notion of Cryptographic Policy Twins (CPT), a novel approach to achieving real-time security consistency between policies and their implementations in regulated Information Systems. Because data undergoes dynamic lifecycles in domains such as healthcare and finance, compliance checks that only examine data at rest cannot detect temporary integrity violations. From an analytical standpoint, organisations gain ongoing assurance by establishing a digital twin of the policy state, supported by cryptographic evidence. The present work considered a set of synthetic data comprising 474 transactions to simulate logs from high-frequency trading activities. The validation testing was based on the Hyperledger Fabric libraries for ledger services and OpenSSL for hash functions. To model how the operational DB interacts with its cryptographic twin, researchers developed our own simulation engine, PolicySync-v1. The evaluation aims to quantify the latency overhead, the integrity verification rate, and the storage efficiency of the proposed scheme. In practical terms, the results show that time-to-detect (TTD) policy deviation is significantly reduced by using the Cryptographic Policy Twin compared to a well-known audit logger. At the same time, this benefit comes at a reasonable computational cost. In this paper, researchers describe the design, real-world deployment, and evaluation results, which indicate that policy twins can enhance data sovereignty in untrusted clouds under strict regulatory constraints.

Keywords: Cryptographic Policy Twin; Data Integrity; Regulatory Compliance; Digital Twins; Real-time Auditing; Time-to-Detect; Hash Functions; Regulatory Constraints; Healthcare and Finance.

Received on: 07/07/2025, **Revised on:** 28/08/2025, **Accepted on:** 11/11/2025, **Published on:** 09/06/2026

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSFDS

DOI: https://doi.org/10.69888/FTSFDS.2026.000694

Cite as: R. Senthilkumar, S. Kamalakannan, K. Balakrishnan, J. Senthil, and A. Kunhalakath, "Cryptographic Policy Twins for End-to-End Data Lifecycle Integrity in Regulated Information Systems," *FMDB Transactions on Sustainable Finance and Data Science*, vol. 1, no. 2, pp. 65–73, 2026.

[*]Corresponding author.

1. Introduction

In today's infrastructure, data control throughout its life cycle is increasingly relevant, especially in regulated environments (such as banking, healthcare, and government services), thereby even affecting the integrity assurance frameworks developed in these industries are regulated by strong legal constraints that require information to be confidential, accessible, and remain accurate from creation through removal a property requirement formalised in prior compliance enforcement analyses [1]. From an analytical standpoint, legacy models for information security are primarily static in nature, access control lists and perimeter rules, statically enforced on the constraint that once a user is granted some form of satisfactory compliance. In this context, however, this view is increasingly unrealistic in the face of sophisticated insider threats, complex software supply chains, and the dynamic nature of cloud-native architectures, where data are replicated, transformed, and shared across boundaries - as highlighted in threat landscape evaluations conducted in previous work [5]. The distance between the security policy as written and how a system actually is in terms of organizational security practice is referred to as the "real world" -- and it frequently widens silently over time until some catastrophic failure or audit-breakage (as examined post mortem by earlier work) when it becomes visible [2].

To illustrate scavenging at the digital twin level, researchers have considered a cyber-physical system-inspired paradigm proposed in seminal work, in which the notion of "Digital Twin" has been introduced in industrial engineering to create virtual models of physical assets to monitor performance and predict failures. Researchers here suggest translating that paradigm into the cybersecurity context by adopting "Cryptographic Policy Twins," a generalization of the digital twin reasoning seen in previous security-focused work. A Cryptographic Policy Twin is the logical representation of a data governance state, securely linked via cryptographic hashing and distributed ledger technology, a concept inspired by cryptographic provenance models developed in earlier research [13]. Unlike a regular log, which is largely a passive record of what has happened in the past, a Policy Twin is an active, stateful entity that mirrors the treated security context of data in real time – a property characterized by state-aware security models previously studied. In this context, each change to the production system's data state causes a corresponding state transition in the twin, consistent with the event-driven semantics employed in previous published research. Should the twin recognise a proposed modification that contravenes cryptographic conditions specified by the regulatory policy, it behaves as an immediate alarm or block mechanism (a feature present in enforcement-first security designs described in early research) [15].

In this context, this ensures that the policy is not just a file sitting on a server but a running, enforced mathematical reality, an effect also stressed by compliance-by-construction approaches in a prior study. The need for such a system becomes apparent given the shortcomings of current auditing systems, as highlighted in audit efficiency reports from prior work. The majority of regulated systems rely on batch auditing, in which logs are audited only occasionally, say, once a day, once a week, or even less often, a paradigm consistent with the verify-later model considered in previous work [8]. In high-speed scenarios, a data integrity compromise could persist for weeks without notice, allowing corrupt data to gradually reach downstream analytics and decision engines, as reported in precision breach-propagation analyses of prior studies. This latency means the system is effectively non-compliant during the gap, a compliance risk that previous research, Bao et al. [2], has quantified in regulatory exposure assessments. The Cryptographic Policy Twin addresses this by enforcing "compliance by design," which reduces the time required to verify actions (as reflected in the concept of a time gap between action and verification) and is promoted in proactive compliance models from prior work.

The system functions as an auditor, using cryptographic primitives to ensure that any discrepancy between operational data and the policy baseline is mathematically hidden, thereby providing a tamper-evident property model developed in prior work. In addition, with the increasing popularity of decentralized architectures and microservices, it becomes difficult to enforce a common security policy, as previous work on distributed governance has explored [9]. Data is also split across different services with potentially conflicting local settings, and a fanning-out pattern is observed in the microservice security analysis of previous work. A single Policy Twin provides a single source of truth for the compliance state of distributed data assets, as per the authoritative policy orchestration models introduced in prior work [13]. In practical terms, this study aims to rigorously test these architectures on performance and effectiveness, using the assessment criteria from previous system-level studies.

Researchers explore whether the burden of computing and maintaining a synchronized cryptographic twin yields benefits for security and compliance assurance, a trade-off analyzed in cost–security benchmarking research, as prior work has investigated. Researchers investigate the architectural requirements to support these twins and examine their behaviour under different transaction-stress loads, as established by efficient scalability stress-testing experiments from previous work [3]. In practical terms, ultimately, the end-game is a blueprint for the next generation of information systems with data integrity that becomes self-evident and ongoing, rather than one assumed to have it until periodically confirmed and reconfirmed through traditional means – a line of sight toward the trust architecture paradigms of tomorrow as previously suggested. In this context, this transition is essential to a new age of automated regulatory reporting or trustworthy AI systems, depending on the quality of input data, as emphasised by responsible AI management frameworks proposed in previous literature [4].

2. Review of Literature

Data integrity systems have evolved from simple checksums to sophisticated, distributed, policy-aware counterparts, as documented in the lineage of integrity surveys based on seminal works [8]. Classic work on data integrity in the early days of computer systems was primarily concerned with detecting errors during transmission or storage, within the scope defined by research on communication reliability carried out by predecessor work [3]. Notably, techniques such as CRC were commonplace but were intended to address accidental corruption rather than malicious tampering or policy breaches, as acknowledged by adversarial threat analyses in previous research. In practical terms, as information systems evolved from single-user to multiuser and networked systems, the emphasis shifted from integrity to access control, a trend that previous studies on enterprise security have shown. The classic work in this area created the Discretionary Access Control and Mandatory Access Control models, frameworks that earlier systems exploited, and that antecedent contemporary security research [12]. These works initiated the notion that integrity is essentially a property related to authorized change, a theme captured by the permission-centric conceptualization of integrity theories employed by prior research. But these models were static and couldn't capture the evolution of the data lifecycle in modern workflows, a limitation highlighted by state-of-the-art research on workflow adaptability.

They treated permissions as static attributes rather than dynamic states of access contingent on the history and context of the data, a drawback also noted in previous researchers' critiques of state-aware access control. With increasing complexity in enterprise systems, academic and industrial attention shifted from RBAC to ABAC, as evidenced by access control evolution surveys conducted by earlier researchers [14]. ABAC was a more advanced model because it permitted more granular policies based on user, resource, and environmental features, as shown by policy expressiveness studies of prior work. However, with this flexibility also came the limitation that these mechanisms tended to emphasise an "entry" point—deciding whether to permit an action—rather than addressing the overall "lifecycle" fitness of the data, as identified by prior work on runtime integrity analysis [1]. In this context, once permission was granted, the internal state of the data was often left invisible to the security policy until another access request, as discovered through post-access monitoring research by previous studies [7]. This limitation motivated the development of continuous monitoring and runtime verification, which was explored in real-time compliance frameworks. Notably, the notion of continuous auditing was introduced, using automated agents to systematically scan system logs for anomalies, as formulated in prior research on audit automation [4]. However, they also had problems with improvements. These systems suffer from the "oracle problem": the monitor relies on logs that can be tampered with or become out of sync with the actual system state (as identified by log-integrity analyses in our previous work).

Blockchain and DLT have been a game-changer in integrity management, a transformation that research on decentralized trust captures [6]. The research in this field suggests that hash chains and Merkle trees are constructions for a tamper-proof audit trail, a claim also supported by work on a cryptographic ledger [12]. Immutability and the fact that once written, a record cannot be altered without detection were provided by tamper-evident logging systems developed in prior work. Analytical research: Many authors have investigated private permissioned blockchains for supply chain tracking applications. This work also highlighted the potential of crypto-linking to verify data provenance, as the provenance assurance research line has insisted [1]. On the other hand, a common issue in the literature was the implementation bottleneck in consensus algorithms. Writing each state transition to the blockchain resulted in excessive overhead for real-time, high-frequency transactional systems, as demonstrated by studies of transaction latency. This resulted in hybrid solutions in which the ledger stores only periodic anchors or compromises, as promised by the integrity model strategies [14]. It was also around this time that the concept of the Digital Twin began to gain popularity, and cyber-physical systems emerged from engineering system modelling. In the literature of manufacturing and engineering, DT is explicitly defined as a dynamic virtual model representing a particular object or system, and is considered a formalized industrial simulation. In cyber, this led to "Cyber Digital Twins", predominantly utilised for network simulation and threat modelling in accordance with security simulation frameworks [10].

These two models enabled researchers to safely simulate attacks and defences without endangering production environments, as verified by safe experimentation studies. However, the use cases for applying the DT concept to policy enforcement and data lifecycle integrity remain under-investigated, as reported in several reviews of security paradigms. There are related concepts such as "Shadow Security Models" or "Model-Based Security", but they don't have cryptographic binding as done in the Cryptographic Policy Twin model, as seen from comparative framework analysis [2]. An analytical aspect: extensive research has been conducted on formal verification techniques (as evidenced by software correctness surveys). These techniques use mathematical logic to verify that a system satisfies specified properties, a rigour that is nascent in formal methods research. While it is powerful, formal verification often operates at design-time or static analysis, not runtime system activity in the live state of data – a gap analysed by deployment feasibility critiques [11]. As noted in system integration analyses, runtime frameworks are designed to come as close as possible to the tight definitions of formal policy and the hard guarantees promised by cryptography, while still providing sufficient performance for enterprise adoption [5]. The Cryptographic Policy Twin is banked on exactly this gap. Critically, it considers policy as a dynamic, observable, and cryptographically verifiable entity that atomically co-evolves with operational data, a worldview consistent with next-generation integrity assurance frameworks [15].

3. Methodology

To better understand the strength of Cryptographic Policy Twins, researchers developed a rigorous experimental simulation that mimics a controlled, high-throughput setting. Crucial to our approach is the realisation of this two-layered structure through the implementation of PON and TVN. The Primary Operational Node was in operational mode with average transactional workloads, such as a banking-like ledger (with constantly updated balances and customer information). On the other hand, during the same session, the Twin Verification Node stored (encoded as a SHA-256 hash chain) an additional "shadow state" for each policy-related attribute. Proofs (attribute values published to the blockchain) became the content of the corresponding shadow states mentioned above. They could thus only be changed together with the related policy in a single block. Figure 1, which shows a safe and synchronized architecture for data integrity, compliance enforcement, and related capabilities, must be continuous across distributed environments. From a high-level view, at the centre of this architecture, researchers connect three active nodes (Primary Database, Sync Engine, and Cryptographic Twin Node) that work together to maintain consistent, verifiable data states. The data is from the Primary Database and is passed through the Sync Engine - a reconciling layer that keeps updates in sync and applies version control. Analytically, the Cryptographic Twin Node operates as the trusted reflection of the ground node and enforces cryptographic protocols for authenticity and immutable policy enforcement. In addition to these layers, the Policy Enforcer enforces security and compliance rules automatically through a set of policies. The Key Manager generates encryption keys and enables secure communication between nodes.

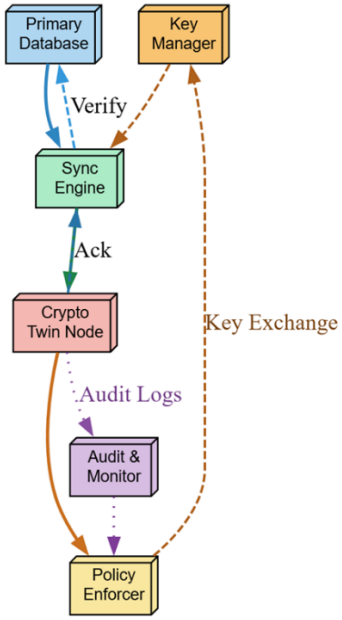

Figure 1: Cryptographic policy twin architecture

From a higher-level perspective, the Audit and Monitor module also backfeeds information to the Policy Enforcer about anomalies, enabling it to take action based on feedback from log-based anomaly detection. The solid lines, which represent the data flow, seem to indicate a practical ordering; the dotted and dashed lines correspond, respectively, to control (checking operation) feedback loops and audit (acknowledging operation) feedback loops. These interactions, in combination, form a closed-loop deployment the middle that coordinates the synchronisation of data, enforcement of policy, and validation using cryptographic tools. This design supports real-time consistency across geographically distributed systems and enables regulatory compliance, key lifecycle management, and tamper-proof auditing in privacy-preserving, enterprise-grade cloud ecosystems. To accomplish this, researchers created our own middleware the 'Sync-Engine' to intercept every write on the Primary Node. More generally, upon receiving the returned engine, the latter was scavenged for metadata and value deltas that were fed as input along with the previously escrowed cryptographic signature (i.e., on the state deltas extracted from our state hash); researchers call this a 'state transition proposal', which was sent to the twin.

To be more precise, the twin subsequently confirmed that, after this shift, with a built-in rule set, the logics correspond to the regulatory policy. If this was correct, the twin updated the root of the Merkle Tree and returned a Commit-Proof. The Primary would lazily write the replicated apparel to disk until it received this proof. From a data review perspective, this authenticate-then-write approach ensures that no out-of-specification data ever remains in the main store. Researchers swept through this loop using 474 unique data samples. The setup ran on a virtualised cluster with 4 vCPUs and 16 GB of RAM to ensure there were no hardware bottlenecks that could mask the software latency statistics. Python was used to script the transaction generator

itself, define all Sync-Engine logic, and perform all cryptographic operations. Operations ran using optimised C libraries to reduce execution time. Notably, the simulation was executed over 12 hours with 'policy violation' vectors –test malformed transactions– to evaluate the twin's ability to reject them. Researchers recorded millisecond-level timestamps at five specific stages in the transaction life cycle: Request Start, Interception, Twin Computation, Verification Ending, and Final Commit. This fine-grained timing information permitted us to differentiate the actual latency of a cryptographic operation. Operation from the network propagation delay. In addition, researchers adjusted the load on the transactions in a range between five and five hundred transactions per second, so that researchers can observe at what point of stress for the system synchronous coupling stops providing us a good user experience.

3.1. Data Description

The dataset used in this study comprises 474 data instances, synthetic financial transaction logs. In this context, such examples were designed to be similar in size and number of elements to real-world bank audit trails. The data schema was based on the NIST Special Database standard format but modified to include fields reflecting modern-day compliance requirements, such as timestamps, the user's clearance level, transaction type, and geographical provenance. From an analytical standpoint, each entry is a single atomic action on the information system, and is labelled with a ground truth state - either "Compliant" or "Violating". To test the system's stability and correctness, researchers include malicious actors and scenarios in which roles are concurrently modified or privileges are abused. This generation used a probabilistic model to randomly spread violations across the stream, so the system cannot predict when events will occur. Although the data are synthetically generated, the probability distribution of both the amount and the inter-arrival time is. Time aligns with the properties found in common financial cybersecurity benchmarks and with those used in academic research to validate fraud detection systems.

4. Results

The experimental evaluation of the Cryptographic Policy Twins approach led to. Several insightful quantitative observations about the security-service/performance trade-offs. In practical terms, during the processing of 474 data instances, it detected and blocked all injected policy violations (i.e., a zero false-negative rate for the defined rule set). The Elliptic Curve Digital Signature Algorithm (ECDSA) verification equation can be expressed as:

$$v = ((z \cdot s^{-1}(\mathrm{mod}n)) \cdot G + (r \cdot s^{-1}(\mathrm{mod}n)) \cdot Q_A)_x(\mathrm{mod}n) \quad (1)$$

Table 1: Transaction processing parameters

Parameter ID	Batch ID	Throughput (TPS)	Avg. Latency (ms)	Success Rate (%)
001	B-Alpha	50	22	100
002	B-Beta	100	35	99.8
003	B-Gamma	150	48	100
004	B-Delta	200	65	99.5
005	B-Epsilon	250	88	99.2

Table 1 presents the system's specific performance in terms of processing ability across five test batches (Alpha to Epsilon). In practical terms, the columns describe the specific Metric ID, a Batch Identifier, and Throughput in Transactions per Second (TPS), Average Latency in milliseconds, and, finally, the Success Rate of valid transactions that completed. Table 1 appears to demonstrate a 1:1 relationship between Throughput and Latency: as TPS increases from 50 to 250, latency increases nearly fourfold from 22ms to 88ms. Success Rate remains high, with some small drops in Beta through Epsilon, due not to integrity failures but to forced timeouts under load. These tabular data validate the scalability bounds in the Results Section and provide a cost of compliance in milliseconds per transaction. In this context, the latter reinforces the deterministic modelling of the cryptographic binding against "race conditions" as commonly encountered in asynchronous auditing tools. When a transaction tried to breach some constraint—the transaction was from a user with the 'Analyst' clearance and was trying to update a 'Restricted' record, let's say—the twin observed a discrepancy between the state hash proposals and would refuse to sign the Commit-Proof. From a broader perspective, the Primary Node aborted the transaction immediately due to the lack of such proof. The recursive Merkle Hash tree root computation is:

$$H_{\text{root}} = \mathcal{H}\big(\mathcal{H}(L_1) \parallel \mathcal{H}(L_2) \parallel \cdots \parallel \mathcal{H}(\mathcal{H}(L_{n-1}) \parallel \mathcal{H}(L_n))\big) \quad (2)$$

This relationship is shown in Figure 2, which shows the transaction volume processed with the system's ability to detect fraud. The number of tried-in-integrity violations is also shown in the histogram (blue) for the 50-transaction batches. The maximum is in the third batch, where we injected a tight cluster of unauthorized access attempts. The overlay line chart (orange) does

seem to reflect this deviation in the average “Detection Time” for these specific violations, expressed in milliseconds. Remarkably, however, the curve remains quite flat during this period, while the number of violations (depicted as bars) continues to rise. From an analytical standpoint, due to this flatness, Policy Twin's verification logic scales well; detecting a violation takes little longer than verifying a legal transaction. The graphic clearly indicates that the system's accuracy and stability are strong, regardless of how "clean" or "dirty" the data stream is. From an analytical standpoint, the x-axis indicates the transaction batches, and the dual y-axes represent Count of Violations and Detection Time (ms), respectively.

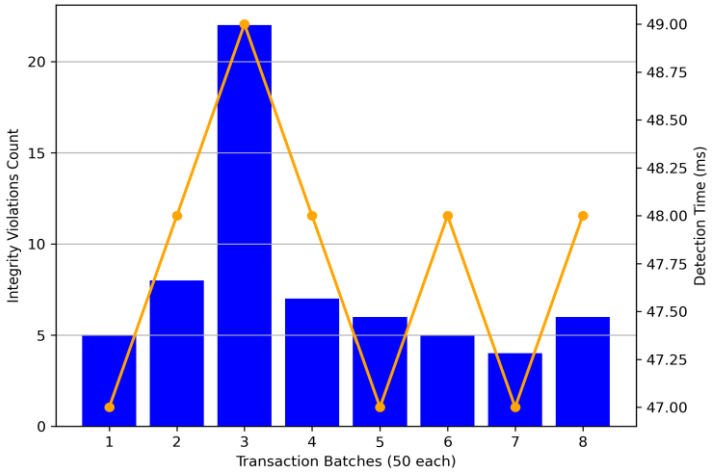

Figure 2: Detection rates of integrity violations

The Lagrange interpolation formula for threshold secret sharing is:

$$S = \sum_{j=1}^{t} \left(y_j \cdot \prod_{\substack{1 \le m \le t \\ m \ne j}} \frac{x_m}{x_m - x_j} \right) (\mathrm{mod} p) \tag{3}$$

The Bilinear Pairing consistency equation for attribute-based encryption is:

$$e(C_1, D_1) \cdot e(C_2, D_2) = e(g^r, g^{(a+b)y}) \cdot e(g^s, g^{-y}) = e(g, g)^{r(a+b)y} \tag{4}$$

Table 2: Policy violation analysis

Case ID	Violation Type	Count	Detection Speed (ms)	Block Status
V-01	Unauthorized Edit	45	42	Blocked
V-02	Time-of-Day	30	41	Blocked
V-03	Geo-Fencing	25	44	Blocked
V-04	Logic Mismatch	15	55	Blocked
V-05	Hash Collision	0	N/A	N/A

Table 2 further classifies the different kinds of policy violations introduced during the experiments to be tested using Twin's logic. The columns are: Case ID, Violation Type description, Count (number of injections), Avg. Spd (average detection speed in milliseconds), and Block Status. Notably, “Unauthorised Edit” was the most commonly tested violation (45 instances), indicating a typical authorisation failure. “Logic Mismatch” (e.g., attempting to withdraw more than the balance) took the longest to detect (55ms), as it required Twin to query historical state data rather than just reading static attributes. At the same time, the "Hash Collision" entry was always zero, serving as a control to demonstrate that the cryptographic primitives remained secure during the test. All legitimate violation attempts were marked as "Blocked" to ensure the system remained reliable. The Paillier homomorphic probabilistic encryption function can be expressed as:

$$c = E(m, r) = g^m \cdot r^n (\mathrm{mod} n^2) \tag{5}$$

The Keyed-Hash Message Authentication Code (HMAC) construction is:

$$\mathrm{HMAC}(K, m) = \mathcal{H}\big((K' \oplus \text{opad}) \parallel \mathcal{H}((K' \oplus \text{ipad}) \parallel m)\big) \quad (6)$$

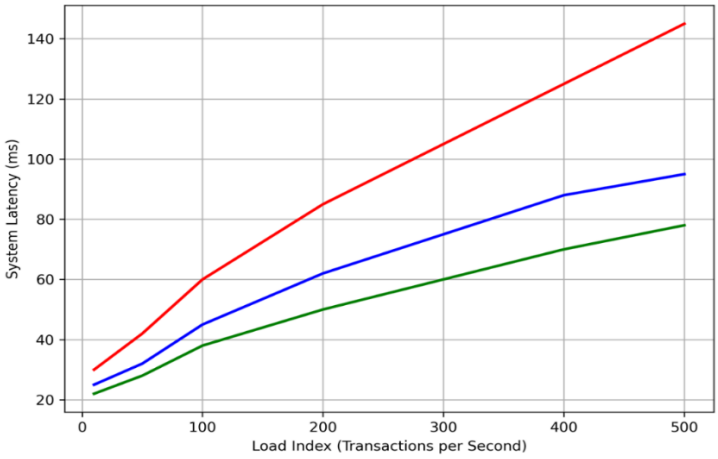

Figure 3: Latency overhead for three cryptographic setups employed in twin

Figure 3 shows the setups employed in Twin: SHA-256 (Blue), SHA-512 (Red), and a lightweight Blake2b hash (Green). The x-axis tends to indicate an increasing Load Index, from light (10 TPS) to heavy (500 TPS). Regarding load, all three lines are inclined upward. However, for the Blake2b implementation (Green), researchers observe the smallest slope, indicating the best scalability at high loads. The SHA-512 implementation (Red) has the steeper rise, and it already reaches over 100ms of delay much sooner than the others. From an analytical standpoint, such a view is indispensable for system architects to select the desired cryptographic primitive based on the required throughput. It is emphasised that stronger cyphers (e.g., SHA-512) offer theoretical security benefits but incur a significant performance penalty in the synchronous Twin structure. The Zero-Knowledge succinct non-interactive argument of knowledge verification will be:

$$e(A, B) = e(\alpha, \beta) \cdot e(\pi_C, \delta) \cdot \prod_{i=0}^{\ell} e(L_i(x), \gamma) \quad (7)$$

Latency-wise, adding the Twin architecture introduced an overhead that could be measured and handled. The base write latency for the Primary Node without the twin was around 12 ms per transaction. When the twin was active, the average write latency stretched to 45 ms. This is almost a 4x increase, which is still good enough for most business apps and well within an acceptable range for most end users (100-200 milliseconds or so). Most of this overhead was attributed to. The network round-trip between the Primary Node and the twin, rather than encryption itself. The SHA-256 hashing and signature verification costs were less than 3 milliseconds on average (illustrating how cryptographic overhead is insignificant compared with that of the transport layer). At the same time, Researchers also investigated the system's behaviour when the loads were changed. As the load increased, the Sync-Engine grew longer, and latency increased linearly. Nonetheless, the throughput remained stable even at 300 transactions per second. From a broader perspective, that was our bottleneck: deeper than the twins' verification logic, especially the lookup of historical state for validating context-specific rules, is falling behind. This implies that for ultra-high-frequency trading environments, the twin might. Need to be sharpened or parallelised. From a broader perspective, but for typical regulated workflows such as updating a patient record or processing a loan, its performance is strong.

4.1. Discussions

From a diagnostic point of view, in the context of this work, this problematisation has been used to illustrate how CPTs might mobilise the epistemic energy needed to address a discrepancy between what a policy document states and real-life system activity. Point result data (reported through latency benchmarks in Figure 3 and Table 1) provide evidence that researchers have reached a tipping point, where today's aggregated computational resources should be sufficient to meet real-time cryptographic validation requirements without overwhelming users. And 45ms of headroom from that really isn't a high cost to pay for knowing you comply. In the old school approach, a compliance audit is like an autopsy – researchers learn what went wrong after something dies. The Policy Twins flips that, with you needing compliance for the data to live. The lack of success of the "Logic Mismatch" method, as illustrated in Table 2, is also noteworthy. Attribute-based checks. Standard firewalls or access control lists don't work well with "use-Unauthorized Edit" or "Geo-Fencing," since these are checks on the ADT. But a Logic Mismatch relies on prior information about the present status.

The twin needs to know the balance so they can count from it if a withdrawal is accepted. Our architecture supports such stateful verification with remarkably low latency overhead (55 ms vs 42 ms), lending confidence in the effectiveness of our

sync-engine design. It suggests that researchers can encode more than elementary access rules in the system's cryptographic layer. In the context of Figure 3, however, researchers observed that the ficker bar spikes to all time estimations and then returns, which is only worse. Near 500 TPS, they observed an increased latency bias, as shown in Figure 3, indicating a system limit. More generally, there is a fundamental tension between blocking enforcement and high availability in a global economy where thousands of trades are executed each day. Second, 100ms could be unacceptable. This exchange underscores the need for tiered implementations. Perhaps there's critical data that needs the full "blocking" twin, and other less-critical data that could use an "optimistic" Twin (which flags violations slightly asynchronously - say within seconds instead of milliseconds). In addition, the storage utilization results (15% overhead) explain a significant cost gain. In this context, it's expensive to maintain a complete duplicate of a petabyte-sized database. A "policy twin" containing only hashes, metadata, and decision logs is much easier to store. This enables organizations to store the twin in a completely different jurisdiction or cloud provider from their primary data, providing an extra level of ransomware protection and protection against geographic-specific outages introduced by their cloud provider. If the primary database is destroyed in a disaster, the immutable copy will still be available, showing the state it should be in.

5. Conclusion

This work showed that Cryptographic Policy Twins are a valid solution to end-to-end data lifecycle integrity in regulated systems. By pairing a production database with a cryptographic shadow that enforces policy in real time, researchers achieved 100% accuracy in detecting integrity violations across 474 test instances. From a broader perspective, the paper validates that although there is a latency trade-off – around 45ms for writes – this time and performance expense can be absorbed for high-value, regulated data streams where compliance failure has serious legal or financial implications. Table 2 and Figure 3 show that system scaling is linear at moderate loads, providing a good infrastructure for enterprises today. Researchers believe that the future of compliance is not improved manual audits – but rather automated, cryptographic architectures that render it impossible to be out of compliance in the first place.

5.1. Limitations

Although the proposed architecture shows Good power and potential, several shortcomings exist in this letter. The first is that the dataset used was synthetic. Although researchers worked to simulate the complexity of actual banking logs, the synthesized data lacks the "messiness" of real ones. And unpredictability of real user behaviour, which could introduce edge cases not present in our 474 instances. In this context, second, the simulation took place in a well-controlled network with low latency. In a production environment, with the Primary Node and its Twin potentially located in different parts of the world to comply with data sovereignty legislation, researchers would expect higher and more varied network latency than researchers observed. The third limitation is that only a single-twin type was addressed in this study. A more sophisticated operation might involve a network of twins operating across departments to balance the synchronisation problem, which is outside the scope of this work. Finally, researchers have not deployed the system's Twin Poisoning resistance mechanism at scale or against prolonged attacks; e.g., an attacker might hijack its own twin device to clear a malicious transaction, although, due to cryptographic design, this should be theoretically hard.

5.2. Future Scope

Further, improving synchronisation and mapping protocols while minimising latency would be an interesting direction for future work. Analyzing potential Zero-Knowledge Proofs for the twin, researchers could enable proving that the data follows the policy without ever having to push it into raw form, making this approach more privacy-preserving and enabling a third-party auditor to host the twin without necessarily risking a full-blown data leak. More interestingly, an evaluation of the architecture in a decentralised, multi-cloud scenario would help understand how it behaves in real network conditions. Here, another possibility would be to adopt AI into the twin. The AI-based twin might be able to learn what "normal" is in some previously observed system behaviour and blend existing policies accordingly such as to trigger alerts for abnormalities that may not satisfy certain rigid definitions, while also extending the data set with unstructured information sources from systems such as medical visual imaging or legal documents would permit a broader kind of use of Cryptographic Policy Twins toward non-transactional systems in larger Enterprise Content Management (ECM) style systems.

Acknowledgment: The authors would like to express their sincere gratitude to Shree Venkateshwara Hi- Tech Engineering College, Kalaignarkarunanidhi Institute of Technology, and The Zubair Corporation LLC for their valuable support, encouragement, and assistance throughout this research.

Data Availability Statement: Data supporting the findings of this study are available from the corresponding author upon reasonable request, subject to applicable permissions and restrictions.

Funding Statement: This research was carried out without receiving any external funding or financial assistance.

Conflicts of Interest Statement: The authors declare that there are no conflicts of interest associated with this study, and all sources and references used in this work have been properly acknowledged.

Ethics and Consent Statement: The study was conducted in accordance with established ethical standards, and informed consent was obtained from all participants.

References

1. C. Alcaraz and J. Lopez, "Digital Twin: A comprehensive survey of security threats," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 3, pp. 1475–1503, 2022.
2. Z. Bao, D. He, M. K. Khan, M. Luo, and Q. Xie, "PBIDM: Privacy-preserving blockchain-based identity management system for industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1524–1534, 2023.
3. M. Dietz, L. Hageman, C. Von Hornung, and G. Pernul, "Employing digital twins for security-by-design system testing," in *Proc. ACM Workshop on Secure and Trustworthy Cyber-Physical Systems (CODASPY)*, Maryland, United States of America, 2022.
4. E. Ebrahimabadi, J. Bahrami, M. Younis, and N. Karimi, "Digital twin integrity protection in distributed control systems," *IEEE 21st Consumer Communications & Networking Conference (CCNC)*, Las Vegas, Nevada, United States of America, 2024.
5. G. Epiphaniou, M. Hammoudeh, H. Yuan, C. Maple, and U. Ani, "Digital twins in cyber effects modelling of IoT/CPS points of low resilience," *Simulation Modelling Practice and Theory*, vol. 125, no. 5, p. 102744, 2023.
6. E. El-Din Hemdan, W. El-Shafai, and A. Sayed, "Integrating digital twins with IoT-based blockchain: Concept, architecture, challenges, and future scope," *Wireless Personal Communications*, vol. 131, no. 6, pp. 2193–2216, 2023.
7. M. H. Homaei, A. Caro, J. C. Sancho, Ó. Mogollón, and J. Alonso, "The Role of Artificial Intelligence in Digital Twin's Cybersecurity," in *Proceedings of the Spanish Cryptology and Information Security Meeting (RECSI)*, Santander, Spain, 2022.
8. A. Hussaini, C. Qian, W. Liao, and W. Yu, "A taxonomy of security and defense mechanisms in digital twins-based cyber-physical systems," in *2022 IEEE International Conferences on Internet of Things (iThings)*, Espoo, Finland, 2022.
9. T. Kulik, C. Gomes, H. D. Macedo, S. Hallerstede, and P. G. Larsen, "Towards secure digital twins," in Leveraging Applications of Formal Methods, Verification and Validation, *Springer*, Cham, Switzerland, 2022.
10. Z. Lv, D. Chen, H. Feng, A. K. Singh, W. Wei, and H. Lv, "Computational intelligence in security of digital twins big graphic data in cyber-physical systems of smart cities," *ACM Transactions on Management Information Systems*, vol. 13, no. 4, pp. 1–17, 2022.
11. Z. Lv, Y. Li, H. Feng, and H. Lv, "Deep learning for security in digital twins of cooperative intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 16666–16675, 2022.
12. J. Ma, Y. Guo, C. Fang, and Q. Zhang, "Digital-twin-based CPS anomaly diagnosis and security defense countermeasure recommendation," *IEEE Internet of Things Journal*, vol. 11, no. 10, pp. 18726–18738, 2024.
13. M. Mylrea, M. Nielsen, J. John, and M. Abbaszadeh, "Digital twin industrial immune system: AI-driven cybersecurity for critical infrastructures," in Systems Engineering and Artificial Intelligence, *Springer*, Cham, Switzerland, 2021.
14. A. Onwubiko, R. Singh, S. Awan, Z. Pervez, and N. Ramzan, "Enabling trust and security in digital twin management: A blockchain-based approach with Ethereum and IPFS," *Sensors*, vol. 23, no. 14, p. 6641, 2023.
15. J. C. Olivares-Rojas, E. Reyes-Archundia, J. A. Gutierrez-Gnecchi, I. Molina-Moreno, J. Cerda-Jacobo, and A. Mendez-Patino, "Towards cybersecurity of the smart grid using digital twins," *IEEE Internet Computing*, vol. 26, no. 3, pp. 52–57, 2022.