© JUL 2019 | IRE Journals | Volume 3 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1725089

A Systematic Review of Transparent Data Encryption (TDE) Performance Overhead in Regulated Enterprise Database Estates

Evans Ighodalo[1], Ayinoluwa Kolawole[2]
[1] *Megatrips.com Ltd, Nigeria*
[2] *Personalised Haven, Nigeria*

Abstract- ***Transparent Data Encryption (TDE) has become the default control that banks, insurers, healthcare providers and public bodies use to meet encryption-at-rest obligations under PCI DSS, HIPAA, the GDPR, the NYDFS Cybersecurity Regulation and the Nigeria Data Protection Regulation, yet its performance cost is reported inconsistently, ranging from vendor estimates of a few percent to practitioner accounts of severe disruption. This paper reports a systematic literature review, conducted in 2019 following the Kitchenham and Charters guidelines, of peer-reviewed studies, standards and vendor documentation covering Oracle Database, Microsoft SQL Server, IBM Db2, MySQL and MongoDB, supported by two contextual seed corpora of 50 regional and 50 international studies. The review finds that steady-state overhead is modest on processors with hardware AES acceleration, because TDE encrypts pages at the I/O boundary and keeps plaintext in the buffer cache, but that overhead rises with I/O-intensive workloads, low cache hit ratios, limited CPU headroom, virtualization that hides AES-NI and the loss of downstream compression. The largest costs arise during lifecycle events, namely initial encryption, backup and restore, key rotation and dependence on external key managers. From these findings the paper develops the TDE Performance Overhead Model, which links regulatory drivers and encryption scope to steady-state and lifecycle overhead through seven testable propositions. The evidence base is dominated by vendor sources with few independent measurements, so the paper concludes with a measurement protocol and deployment recommendations for regulated estates and an agenda for empirical research.***

Index Terms- ***AES-NI, conceptual model, database performance, encryption at rest, enterprise database estates, key management, regulatory compliance, systematic literature review, Transparent data encryption***

I. INTRODUCTION

Enterprise database estates in regulated industries are large, heterogeneous and long-lived. A typical tier-one bank or hospital group operates hundreds to thousands of database instances spanning Oracle Database, Microsoft SQL Server, IBM Db2, MySQL and, increasingly, document stores such as MongoDB. Risk-based assurance research emphasizes that such estates must protect confidentiality without compromising availability, including against insider threats (Akeju et al., 2018; Fadayomi et al., 2019), and regulators have responded with progressively more explicit expectations for data protection (Mbonu et al., 2018; Mbonu, Aliliele, et al., 2019). These estates hold cardholder data, protected health information, personally identifiable information and financial records that regulators expect to be protected not only in transit and through access control but also while stored on disk, tape and cloud object storage.

Transparent Data Encryption has emerged as the dominant answer to this expectation. Introduced by Oracle as column-level encryption in Oracle Database 10g Release 2 and extended to tablespace encryption in 11g, and by Microsoft as database-level encryption in SQL Server 2008, TDE encrypts data files, and usually transaction logs and backups, using symmetric keys managed by the database engine. The term "transparent" refers to the fact that authorized sessions see plaintext without any change to SQL statements, drivers or application code. For organizations under audit pressure this property is decisive: a control can be applied to an existing system in weeks rather than requiring a multi-year application remediation programme, a property that also suits the move toward continuous, automated evidence of control effectiveness (Akomolafe & Agu, 2018; Dosunmu & Ogundele, 2019).

The question that recurs in architecture review boards, capacity planning exercises and vendor negotiations is what TDE costs in performance. Answers in circulation vary widely. Vendor documentation tends to cite small percentages; practitioner blogs and conference presentations report cases of substantially longer backup windows, CPU saturation during initial encryption and unexpected effects on shared resources such as the temporary database. Decision makers in regulated estates therefore lack a consolidated, critically appraised view of the evidence.

© JUL 2019 | IRE Journals | Volume 3 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1725089

A. Research objectives and questions

The objective of the review is to establish what is known, and how reliably, about the performance cost of TDE in regulated enterprise estates, and to organize that knowledge into a model that can guide measurement and deployment decisions. Four research questions structure the review (Table 1).

Table 1. Research questions

ID	Research question	Motivation
RQ 1	What steady-state performance overhead does TDE impose on transactional and analytical database workloads?	Establish the baseline cost that capacity planners must budget for
RQ 2	Which workload, configuration and infrastructure factors moderate that overhead?	Explain the variance between reported figures and identify levers
RQ 3	What operational (lifecycle) costs does TDE impose beyond steady state?	Capture costs such as initial encryption, backup and key rotation that dominate rollout risk
RQ 4	What is the quality and coverage of the available evidence?	Judge how far decision makers can rely on published figures

B. Contributions and structure of the paper

The paper makes four contributions. First, it consolidates the published evidence on TDE performance overhead across platforms into a taxonomy that separates steady-state from lifecycle costs. Second, it identifies the workload, configuration and infrastructure factors that moderate overhead. Third, it proposes a conceptual model, expressed as testable propositions, that explains how regulatory drivers translate into performance outcomes. Fourth, it appraises the quality of the evidence and proposes a reproducible measurement protocol that regulated organizations can apply before estate-wide rollout.

The remainder of the paper is organized as follows. Section 2 describes the research method. Section 3 presents the results and discussion: the literature synthesis, covering TDE architecture, the regulatory context, the evidence for each research question and the adjacent-domain literature; the conceptual model; and the discussion, implications, limitations and directions for future research. Section 4 concludes.

II. RESEARCH METHOD

A. Review design

The review followed the guidelines for systematic literature reviews in software engineering proposed by Kitchenham and Charters (2007), with reporting informed by the PRISMA statement (Moher et al., 2009). Because the topic spans academic research and industrial practice, the protocol deliberately included authoritative grey literature, namely vendor technical documentation and white papers, alongside peer-reviewed sources. This is consistent with multivocal review practice in software engineering, where much relevant knowledge is published outside academic venues (Garousi et al., 2019), and with established guidance on structuring systematic and concept-centric literature reviews (Kitchenham et al., 2009; Webster & Watson, 2002).

B. Search strategy

Searches were conducted across the ACM Digital Library, IEEE Xplore, SpringerLink, ScienceDirect and Google Scholar, and supplemented by targeted searches of vendor documentation portals (Oracle Help Center, Microsoft Docs, IBM Knowledge Center, MySQL and MongoDB manuals). The base search string combined three concept groups:

- ("transparent data encryption" OR "TDE" OR "tablespace encryption" OR "database encryption" OR "encryption at rest" OR "native encryption")
- AND ("performance" OR "overhead" OR "throughput" OR "latency" OR "benchmark" OR "CPU")
- AND ("database" OR "DBMS" OR "SQL Server" OR "Oracle" OR "Db2" OR "MySQL" OR "MongoDB")

The search window covered publications from 2002, the period of the earliest academic work on encrypted database processing, to mid-2019. Backward and forward snowballing was applied to all included academic sources, following established snowballing guidelines (Wohlin, 2014).

In addition to the database search, a seed-based strategy was used to capture adjacent-domain literature. A consolidated reference list of peer studies, merged and deduplicated across several research portfolios and organized by subject area, was screened by title for relevance to encryption, data protection, security architecture, compliance, audit, data infrastructure and IT operations. Consistent with the 2019 review date, only entries published in 2019 or earlier were eligible. Entries judged related or semi-related were retained as a contextual seed corpus of 50 studies. Because these studies address the environment in which TDE operates rather than measuring TDE

© JUL 2019 | IRE Journals | Volume 3 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1725089

itself, they were synthesized thematically (Section 3.9) and were not scored against the measurement-oriented quality criteria in Section 2.4.

To balance this regional corpus, a second, international seed corpus of 50 studies published up to 2019 was assembled from the reference lists of included sources and from targeted searches of the ACM, IEEE, USENIX, Springer and Elsevier literature and of international standards bodies. Studies were retained when they addressed encrypted data management, storage and file-system encryption, block cipher performance, key management, or the database, virtualization and storage mechanisms through which TDE overhead arises. This corpus is reviewed in Section 3.9.7.

C. Inclusion and exclusion criteria

Table 2. Inclusion and exclusion criteria

Criterion type	Criterion
Inclusion I1	Source reports measured, estimated or architecturally derived performance effects of engine-level encryption at rest
Inclusion I2	Source describes TDE mechanisms in sufficient detail to reason about where overhead arises (for example, key hierarchy, I/O path, interaction with compression or backup)
Inclusion I3	Source is a peer-reviewed publication, a standards document, or primary vendor technical documentation
Exclusion E1	Marketing material without technical substance
Exclusion E2	Sources concerned solely with encryption in transit (TLS) or with application-layer encryption without engine-level comparison
Exclusion E3	Duplicate or superseded versions of the same documentation (the latest 2019 version was retained)
Exclusion E4	Anonymous forum posts and unattributed benchmark claims that could not be traced to a reproducible description

D. Quality assessment and synthesis

Each included source was assessed against five quality criteria adapted from Kitchenham and Charters (2007): (Q1) clarity of the objective; (Q2) description of the experimental environment, including hardware and AES acceleration; (Q3) description of the workload; (Q4) presence of an unencrypted baseline; and (Q5) independence from the vendor of the product measured. Each criterion was scored 1 (yes), 0.5 (partly) or 0 (no). Because the included studies used heterogeneous workloads, metrics and platforms, and because most did not report variance, statistical meta-analysis was not feasible. A narrative and thematic synthesis was therefore performed, coding each finding against an overhead taxonomy developed iteratively during extraction.

E. Corpus profile

The final corpus comprises the sources listed in Table 3, grouped by type, together with the seed corpus described in Section 3.9. Standards and regulatory texts were used to frame requirements and cryptographic assumptions; vendor documentation provided mechanism descriptions and the bulk of published overhead characterizations; and academic sources provided architectural analysis of encryption placement and its cost.

Table 3. Evidence profile of included sources

Source	Type	Contribution to the review	Quality notes
Microsoft (2019a, 2019b)	Vendor documentation	Mechanism of SQL Server TDE; stated estimated overhead in the low single-digit percentage range; tempdb and backup behaviour; 2019 scan suspend and resume	Clear mechanism; workload and hardware for estimate not specified; not independent
Oracle (2012, 2019)	Vendor documentation and white paper	Tablespace versus column encryption; buffer cache behaviour; compression ordering; hardware acceleration; online and offline conversion	Detailed mechanism; performance claims qualitative; not independent

Source	Type	Contribution to the review	Quality notes
IBM (2019)	Vendor documentation	Db2 native encryption mechanism; key manager integration	Mechanism only; little quantitative data
Oracle MySQL (2019)	Vendor documentation	InnoDB tablespace encryption and keyring architecture	Mechanism only
MongoDB (2019)	Vendor documentation	Encrypted Storage Engine integration with WiredTiger	Mechanism; general overhead discussion
Gueron (2010)	Industry technical paper	Performance of AES-NI relative to software AES	Detailed and reproducible at the primitive level; not database specific
Iyer et al. (2004)	Peer-reviewed	Framework and cost analysis for storage security inside an RDBMS, including page-level encryption	Explicit baseline; prototype rather than commercial engine
Shmueli et al. (2009); Elovici et al. (2004)	Peer-reviewed	Design considerations for encryption placement, index encryption and their performance implications	Analytical; limited measurement
Hacigumus et al. (2002); Popa et al. (2011); Arasu et al. (2013); Bouganim and Pucheral (2002)	Peer-reviewed	Comparative designs (query over encrypted data, trusted hardware) that contextualize the low cost of I/O-boundary encryption	Strong methodology; address different threat models
Regional adjacent-domain seed corpus (50 studies, 2017 to 2019; Sections 3.9.1 to 3.9.6)	Peer studies (journals, repositories)	Governance, architecture, workload and operational context for TDE deployment	Contextual; mostly conceptual frameworks and reviews; not TDE measurements
International seed corpus (50 studies, 1981 to 2019; Section 3.9.7)	Peer-reviewed papers, standards, technical reports	Encrypted query processing, storage encryption, cipher performance, key management, buffer, virtualization and deduplication mechanisms	Strong methodology; mostly measures related systems rather than commercial TDE
NIST (2001); Dworkin (2001); Scarfone et al. (2007); Barker (2016)	Standards	Cipher, mode, storage encryption and key management assumptions	Authoritative; not performance studies

A striking feature of the corpus is the scarcity of independent, peer-reviewed measurements of commercial TDE implementations under standardized workloads such as TPC-C or TPC-H (Transaction Processing Performance Council, 2010, 2018). Vendor licence terms that historically restricted publication of benchmark results without consent are a plausible contributor to this gap, as is the difficulty of reproducing enterprise-scale storage configurations in academic laboratories.

III. RESULTS AND DISCUSSION

This section presents the results of the review and discusses them. The literature synthesis comes first: Sections 3.1 to 3.4 establish how TDE works, how it is implemented on the principal platforms and why regulated organizations deploy it; Sections 3.5 to 3.8 report the evidence for each research question; and Section 3.9 situates these findings within the adjacent-domain seed literature. Section 3.10 develops a conceptual model from the synthesis. Section 3.11 discusses the findings, Section 3.12 sets out implications for practitioners, auditors and researchers, Section 3.13 states the limitations of the review and Section 3.14 identifies directions for future research.

A. Encryption at rest in the database stack

Encryption of stored data can be applied at several layers of the stack: within the application, within the database engine at column or cell level, within the database engine at page or tablespace level, at the operating system or file-system layer, or at the storage device itself through self-encrypting drives or array-

based encryption (Scarfone, Souppaya and Sexton, 2007). Each layer offers a different trade-off between the threats it addresses, the transparency it offers to applications and the computational cost it imposes. Application-level and column-level schemes protect against a wider range of insider threats, including privileged database administrators in some designs, but impose query-processing restrictions that academic work has studied extensively (Hacigumus et al., 2002; Popa et al., 2011; Arasu et al., 2013). Storage-layer encryption is nearly free in CPU terms but protects only against physical media loss (Dworkin, 2010; IEEE, 2008).

TDE sits between these extremes. It is implemented inside the database engine, so it protects database files, redo or transaction logs and backups wherever they are copied, yet it encrypts at the granularity of a page or block as data moves between the buffer cache and storage. This design choice is the principal determinant of its performance profile and is examined in detail below.

B. TDE architecture

Although implementations differ, the major platforms share a common architecture with three elements.

- A two-tier or three-tier key hierarchy. Data is encrypted with a data encryption key (DEK), called the database encryption key in SQL Server and the tablespace key in Oracle. The DEK is itself encrypted by a master key held outside the data files, in a certificate protected by the service master key, an Oracle keystore (wallet), an external key manager reached over KMIP, or an HSM.
- Encryption at the I/O boundary. Pages are encrypted when written from the buffer pool to disk and decrypted when read from disk into memory. Pages resident in the buffer cache are held in plaintext, so logical reads of cached data incur no cryptographic cost.
- Symmetric block ciphers, predominantly AES with 128-bit or 256-bit keys (NIST, 2001), in modes chosen by the vendor. Some platforms also support legacy algorithms such as Triple DES for compatibility.

Because cryptographic work is incurred only on physical I/O, the arithmetic of overhead is straightforward in principle: the additional CPU time is proportional to the volume of pages physically read and written, multiplied by the per-byte cost of AES on the host processor. The introduction of AES New Instructions (AES-NI) in x86 processors from 2010 onward reduced that per-byte cost substantially compared with software implementations (Gueron, 2010), and comparable acceleration exists on SPARC and POWER processors. The practical consequence is that, on current hardware, the CPU cost of encrypting a page is small relative to the latency of reading it from storage (Akdemir et al., 2010; Käsper & Schwabe, 2009).

C. Platform implementations

Table 4 summarizes the TDE implementations of the principal enterprise platforms as documented in 2019.

Table 4. TDE implementations on major enterprise database platforms (as documented in 2019)

Platform	Introduced	Granularity	Scope of protection	Notable performance-relevant behaviour
Oracle Database (Advanced Security option)	10gR2 (column), 11gR1 (tablespace)	Column or tablespace	Data files, redo, undo, temp, RMAN backups (configurable)	Tablespace encryption decrypts into buffer cache; compression applied before encryption; hardware crypto acceleration used when available; online conversion added in 12.2
Microsoft SQL Server (Enterprise; Standard from 2019)	2008	Database	Data and log files, backups, tempdb	tempdb is encrypted when any user database uses TDE; backup compression with TDE supported from 2016 with large MAXTRANSFERSIZE; initial scan can be suspended and resumed from 2019
IBM Db2 (native encryption)	10.5 FP5	Database	Data, logs, backups	Uses keystore or external KMIP manager; leverages hardware acceleration on POWER and x86

Platform	Introduced	Granularity	Scope of protection	Notable performance-relevant behaviour
MySQL InnoDB (data-at-rest encryption)	5.7.11	Tablespace	Tablespaces; redo and undo from 8.0	Keyring plugin manages master key; encryption at page level on I/O
MongoDB Enterprise (Encrypted Storage Engine)	3.2	Database / storage engine	WiredTiger data files	Integrated with WiredTiger; KMIP key management; overhead tied to cache eviction and I/O
PostgreSQL	Not native in 2019	n/a	n/a	Community proposals under discussion; commercial forks and file-system encryption used instead

Sources: Oracle (2019); Microsoft (2019a, 2019b); IBM (2019); Oracle MySQL (2019); MongoDB (2019).

D. Regulatory drivers

Regulatory frameworks rarely mandate TDE by name. Instead they require that stored sensitive data be rendered unreadable, or they treat encryption as an addressable or risk-based safeguard. Table 5 maps the principal frameworks relevant to enterprise estates to their encryption-at-rest expectations. Two features of this landscape shape the performance question. First, several frameworks create strong incentives to encrypt broadly rather than selectively: breach-notification safe harbours under HIPAA and GDPR attach to encrypted data, and auditors frequently prefer whole-database controls whose coverage is easy to evidence. Second, compliance deadlines often compress rollout timelines, so organizations encrypt large estates quickly and encounter lifecycle overheads, such as the initial encryption scan, at scale. Sector-specific studies of insurance, banking and healthcare compliance illustrate how these pressures play out in practice (Akomolafe & Agu, 2019a; Akomolafe & Agu, 2019c; Badmus et al., 2019b).

Table 5. Regulatory drivers for encryption at rest in enterprise database estates

Framework	Relevant provision	Encryption expectation	Implication for TDE
PCI DSS v3.2.1 (2018)	Requirement 3.4; 3.5 to 3.6 (key management)	Render stored PAN unreadable using strong cryptography with documented key management	TDE commonly accepted for file-level protection; key custody and separation of duties scrutinized
HIPAA Security Rule	45 CFR 164.312(a)(2)(iv) and (e)(2)(ii)	Encryption is an addressable implementation specification	Encrypted PHI qualifies for breach-notification safe harbour under HHS guidance
EU GDPR (2016/679)	Article 32; Article 34(3)(a)	Encryption cited as an appropriate technical measure	Encryption can remove the obligation to notify data subjects of a breach
NYDFS 23 NYCRR 500 (2017)	Section 500.15	Encrypt nonpublic information at rest or apply approved compensating controls	Drives estate-wide encryption programmes in US financial services
Nigeria Data Protection Regulation (2019)	Article 2.6	Appropriate security measures including protection against unauthorized access	Encourages encryption as a demonstrable safeguard for personal data

E. Steady-state overhead (RQ1)

Across the corpus, steady-state TDE overhead on contemporary hardware is consistently characterized as modest. Microsoft documentation for SQL Server has long stated that the performance impact of TDE is low and has given an estimate in the region of three to five percent, while noting that the figure depends on workload (Microsoft, 2019a). Oracle documentation and its best-practice white paper describe tablespace encryption overhead as small, particularly where hardware cryptographic acceleration is available, and emphasize that because blocks are decrypted once into the buffer cache, repeated access to cached data carries no cryptographic cost (Oracle, 2012, 2019). Db2,

MySQL and MongoDB documentation make qualitatively similar claims without precise figures.

These characterizations are consistent with first-principles analysis. AES-NI reduces AES throughput cost to the order of a few processor cycles per byte in parallelizable modes (Gueron, 2010). For a database page of 8 KB, the CPU time to encrypt or decrypt is therefore measured in microseconds, whereas a physical read from spinning disk is measured in milliseconds and from enterprise flash in tens to hundreds of microseconds. Iyer et al. (2004) similarly found in a prototype RDBMS that page-level encryption imposed considerably lower cost than finer-grained schemes because it amortizes cipher setup across whole pages and avoids per-value operations during query processing.

The synthesis therefore supports the following finding: for workloads in which the working set is largely cached and the host provides hardware AES acceleration, steady-state TDE overhead is expected to be in the low single-digit percentage range of CPU and is often difficult to distinguish from run-to-run variance. The finding must, however, be qualified by the weak independence of the evidence, as discussed in Section 3.8.

A second finding concerns the comparison with column-level encryption. Oracle documentation and academic analyses agree that column-level encryption can impose substantially larger overhead in some query patterns, because values are decrypted per row on access, indexes on encrypted columns support only equality predicates in some implementations, and the optimizer may be prevented from using range scans (Oracle, 2019; Shmueli et al., 2009; Elovici et al., 2004). Where regulators accept file-level protection, tablespace or database-level TDE is therefore generally the lower-overhead choice.

F. Moderating factors (RQ2)

Thematic coding identified six factors that moderate TDE overhead. They are summarized in Table 6 and discussed below.

Table 6. Factors moderating TDE steady-state overhead

Factor	Direction of effect	Mechanism	Evidence strength
Hardware AES acceleration (AES-NI, SPARC, POWER crypto)	Absence increases overhead markedly	Software AES costs many times more cycles per byte than accelerated AES	Strong at primitive level; moderate at database level
Buffer cache hit ratio	Lower hit ratio increases overhead	Only physical I/O incurs encryption or decryption	Strong (mechanism documented by all vendors)
Workload I/O intensity (scan-heavy analytics, batch loads)	Higher I/O increases overhead	Full scans and large sorts drive physical reads and temp writes	Moderate
CPU headroom	Saturated hosts show overhead as latency	Extra cycles compete with query execution	Moderate (practitioner reports)
Virtualization and CPU feature exposure	Masked AES-NI increases overhead	Some hypervisor configurations do not pass AES-NI to guests	Low to moderate
Interaction with compression and deduplication	Can increase storage and I/O	Encrypted data is incompressible to downstream storage and backup appliances	Strong (mechanism)

Hardware acceleration. The single most important determinant in the corpus is whether the processor exposes hardware AES instructions to the database engine. Oracle explicitly documents automatic use of hardware acceleration where available (Oracle, 2019), and Gueron (2010) demonstrates large throughput gains from AES-NI over table-based software AES. Estates retaining older hardware, or running on virtualized hosts where CPU feature masking hides AES-NI to preserve migration compatibility, should expect materially higher overhead than vendor estimates suggest.

Cache residency and I/O intensity. Because decryption occurs once per physical read, overhead scales with the ratio of physical to logical I/O. OLTP systems with well-sized buffer pools and small hot working sets incur little overhead. Data warehouses, reporting replicas and batch ETL processes that scan large tables, spill sorts and hash joins to temporary storage, or bulk-load data will incur proportionally more. In SQL Server, the encryption of tempdb whenever any user database on the instance has TDE enabled extends this cost to unencrypted databases that share the instance (Microsoft, 2019a), a consolidation side effect that is frequently overlooked. Scan-heavy analytical workloads of this kind are common in financial risk evaluation, executive decision support and laboratory information systems (Abdulrazaq et al., 2019; Akomolafe & Agu, 2019b; Lawal & Oduleye, 2019b), and IT operations maturity models and cloud database optimization models can be adapted to

forecast their encryption overhead (Anyanwu, 2019; Odejobi et al., 2019).

Downstream storage efficiency. Well-designed ciphertext is statistically indistinguishable from random data. Encryption therefore defeats compression and deduplication performed below the database engine, by storage arrays, backup appliances or WAN optimizers. This does not slow the database directly, but it can increase storage consumption, backup volumes and replication bandwidth, which in turn lengthens backup and restore windows. Engines that compress before encrypting, as Oracle does for compressed tablespaces (Oracle, 2019), preserve the benefit of database-level compression.

G. Lifecycle and operational overhead (RQ3)

The corpus indicates that, for many regulated estates, lifecycle events rather than steady-state processing account for the most significant performance and availability risk. Table 7 summarizes the lifecycle costs identified.

Table 7. Lifecycle overheads of TDE and documented mitigations

Lifecycle event	Nature of overhead	Platform notes	Mitigations
Initial encryption of existing data	Full read and rewrite of every page; sustained CPU and I/O load; log generation	SQL Server background scan, suspendable and resumable from 2019; Oracle offline conversion or online conversion from 12.2	Schedule in low-load windows; throttle; encrypt standbys first and switch over; test duration on a clone
Backup	Native backup compression ineffective or reduced; larger backup sets; appliance deduplication lost	SQL Server 2016 and later support compression with TDE when MAXTRANSFERSIZE exceeds 64 KB; Oracle RMAN compresses before encryption	Use engine-level compression; revisit appliance capacity plans; validate backup window SLAs
Restore and disaster recovery	Certificate or keystore must be available on target; failed restores if keys absent	All platforms	Escrow keys; rehearse restores; include key material in DR runbooks
Key rotation	Re-encryption of data (DEK rotation) is expensive; master key rotation is cheap	SQL Server and Oracle re-wrap DEK on master key rotation without rewriting data	Rotate master keys on policy schedule; reserve DEK rotation for compromise scenarios
External key manager or HSM	Latency at startup, database open and key retrieval; availability dependency	Oracle, SQL Server (EKM), Db2, MySQL and MongoDB support external managers via KMIP or vendor providers	Cache keys per vendor guidance; deploy HSM clusters; monitor key manager availability
High availability and replication	Encryption must be consistent across replicas; log shipping and redo carry encrypted data	Always On, Data Guard and similar require keys on all members	Distribute keys before enabling; coordinate rollout across replica sets

Initial encryption. Enabling TDE on an existing database requires every page to be read, encrypted and written back. For multi-terabyte databases this is a sustained, I/O-bound operation that can run for many hours and that competes with production workload. Prior to SQL Server 2019 the background encryption scan could not be paused in a supported manner, so organizations had either to accept the load or to schedule the operation carefully; SQL Server 2019 introduced the ability to suspend and resume the scan (Microsoft, 2019a). Oracle historically required offline conversion or table-by-table online redefinition, with online tablespace conversion introduced in Oracle Database 12c Release 2 (Oracle, 2019). Initial encryption also generates transaction log or redo volume, which affects log shipping and standby synchronization, a consequence of write-ahead logging design (Mohan et al., 1992).

Backup and restore. The interaction between TDE and backup compression produced the most consistent practitioner concern in the corpus. Before SQL Server 2016, native backup compression yielded little benefit for TDE-enabled databases because pages were already encrypted when read by the backup process. Microsoft subsequently enabled compression of TDE databases when a sufficiently large MAXTRANSFERSIZE is specified, by decrypting, compressing and re-encrypting within the backup stream (Microsoft, 2019b). Regardless of engine behaviour, backup appliances that rely on deduplication lose much of their effectiveness for encrypted data, which has capacity and time

implications at the estate level. Research on financial resilience and on condition monitoring of critical infrastructure underlines that such changes to recovery windows must be reflected in tested continuity plans (Adaramola et al., 2018; Akomolafe & Agu, 2019c).

Key management. Key hierarchy design determines the cost of rotation. Because data is encrypted under the DEK and only the DEK is wrapped by the master key, rotating the master key requires only re-encryption of a small key blob, while rotating the DEK requires rewriting the data. Regulated organizations that interpret key rotation policies (for example, PCI DSS Requirement 3.6.4) as requiring DEK rotation can inadvertently impose recurring initial-encryption-scale workloads. NIST guidance on cryptoperiods (Barker, 2016) allows a nuanced approach in which key-encrypting keys and data-encrypting keys have different cryptoperiods. External key managers and HSMs, which auditors often prefer for separation of duties, introduce a dependency on network latency and service availability at database open and key retrieval, though not usually on each page operation, because the DEK is cached in engine memory. Identity and directory governance determines who can reach the key manager at all (Ladapo et al., 2018; Oshoba et al., 2019).

H. Quality and coverage of the evidence (RQ4)

Applying the quality criteria in Section 2.4, the evidence base is characterized by strong mechanism description and weak independent measurement. Vendor documentation scored well on clarity of objective (Q1) and implicitly on mechanism, but poorly on description of the environment (Q2), workload (Q3) and independence (Q5); headline overhead estimates are typically given without the hardware, workload, cache configuration or variance that would allow them to be reproduced. Academic sources scored well on methodology but measured prototypes or alternative designs rather than the commercial engines deployed in regulated estates. No included source reported a controlled comparison of commercial TDE implementations across platforms on a standardized benchmark with an unencrypted baseline and reported variance.

Coverage is also uneven. SQL Server and Oracle are comparatively well documented; Db2, MySQL and MongoDB documentation offers little quantitative guidance; PostgreSQL lacked native TDE in 2019. Lifecycle costs are documented as mechanisms and mitigations rather than measured. Virtualized and cloud-hosted deployments, now common in regulated estates, are almost absent from the measured evidence despite the plausible influence of CPU feature masking and shared storage on overhead. The regional seed corpus is rich in conceptual frameworks and reviews, and the international corpus is rich in rigorous measurement of related systems, yet neither contains controlled measurements of commercial TDE, confirming that the measurement gap extends beyond the core database literature.

I. Positioning within the broader enterprise security and infrastructure literature

TDE is rarely deployed or evaluated in isolation. In regulated estates it sits within a wider control environment that includes data protection governance, identity and infrastructure architecture, platform and IT operations, audit and internal control, and the reliability management of critical infrastructure. To situate the performance question within that environment, the review incorporated two seed corpora published up to 2019: a regional corpus of related and semi-related studies drawn from a consolidated, cross-disciplinary reference list, and an international corpus of studies on encrypted data management, storage encryption, cryptographic performance, key management and database performance (Section 2.2). These works do not, for the most part, measure TDE directly. They are reviewed here because they describe the governance requirements, architectural patterns, workloads and operational practices that determine where TDE overhead falls and how it is experienced. Sections 3.9.1 to 3.9.6 review the regional corpus by theme and Section 3.9.7 reviews the international corpus, citing each study by author and year.

Data protection, security governance and risk assurance

Comparative reviews of data protection regulation and secure cloud implementation across jurisdictions show that encryption is consistently treated as a baseline safeguard, while conceptual models of legal and ethical risk in enterprise data protection governance stress that controls must be documented within a defensible governance system rather than deployed piecemeal (Mbonu et al., 2018; Mbonu, Aliliele, et al., 2019). This matters for performance because the scope of encryption a governance programme mandates, whether selected columns, sensitive databases or the entire estate, largely determines the aggregate overhead an organization absorbs.

Security audit and enterprise risk assessment frameworks for resilient information systems, and work on risk-based cybersecurity assurance and data availability, frame confidentiality controls such as TDE as one part of a broader assurance case in which availability must not be compromised (Dosunmu & Ogundele, 2019; Fadayomi et al., 2019). That framing maps directly onto the lifecycle concerns identified in

this review, where the principal risk of TDE is not compromised confidentiality but reduced availability during initial encryption, backup and recovery. Insider threat classification and risk modeling (Akeju et al., 2018) is a reminder that privileged administrators remain outside the TDE threat model, and studies of enterprise log analytics and automated incident response using SIEM platforms (Mbonu, Iwuanyanwu, et al., 2019) indicate how key access events and encryption state changes can be brought into security monitoring.

Identity, directory and cloud infrastructure

Because TDE protects data at rest but not data served to authenticated sessions, identity and directory controls are essential complements. Secure identity and access management models for distributed and federated systems (Oshoba et al., 2019), together with practitioner studies of Active Directory implementation and of offline security assessment of Active Directory (Ladapo et al., 2018; Ladapo et al., 2019), describe the controls that govern who can reach both the database and the key material that protects it. These sources reinforce the review finding that key custody and access to the key manager are as consequential as the cipher itself.

Work on scalable and secure cloud architectures, energy-efficient virtual machine placement and constraint-based cloud network resource allocation (Ahmed & Odejobi, 2018a; Ahmed & Odejobi, 2018b; Ahmed et al., 2019) bears on a moderating factor identified in Section 3.6: virtualized and consolidated hosts may mask hardware AES acceleration or concentrate encryption load on shared resources. Performance evaluation of multi-tenant platforms under high concurrency and approximation models for cloud-based database optimization (Odejobi & Ahmed, 2018b; Odejobi et al., 2019) offer analytical approaches that can be adapted to forecast how encryption overhead behaves under contention.

Data integration, platform governance and IT operations

Data integration and delivery practices shape the I/O patterns that drive TDE cost. ETL and data integration design for enterprise platforms, systematic reviews of CI/CD pipeline strategy, and governance frameworks for platform management in regulated healthcare environments (Badmus et al., 2018; Badmus et al., 2019a; Badmus et al., 2019b) determine how often and how heavily encrypted stores are scanned, and how encryption-aware configuration is promoted consistently from test to production. Maturity models for standardizing IT support across distributed organizations (Anyanwu, 2019), reviews of digital transformation in operations management (Agboada, Ezeako, Okoli, & Uzondu, 2018) and frameworks that treat standard operating procedures as change infrastructure (Agboada et al., 2019) describe the operational disciplines through which TDE rollouts are scheduled, executed and sustained.

Two further studies are relevant at the margins. Offline-first design for low-connectivity environments (Unamma et al., 2018) highlights architectures in which encrypted local stores must operate without continuous access to central key services, and work on developing globally employable technology talent (Wikiri, 2019) points to the skills base on which correct configuration and operation of encryption controls ultimately depends.

Audit, internal control and financial-sector compliance

Auditors are the principal consumers of evidence that TDE is in place. Conceptual models for technology-enabled internal audit quality and for risk-based internal control in the insurance and banking sectors (Akomolafe & Agu, 2018; Akomolafe & Agu, 2019a), together with data-driven risk evaluation models and integrated governance and compliance strategies for financial institutions (Akomolafe & Agu, 2019b; Akomolafe & Agu, 2019c), indicate that encryption controls must be demonstrable, consistently scoped and traceable to policy. Reviews of capital adequacy frameworks (Mensah et al., 2019) and of strategic accounting systems for financial transparency and regulatory compliance (Bola-Sadipe et al., 2019) describe the reporting obligations that make data integrity and availability in financial databases non-negotiable.

Related work on budget compliance and statutory reporting, reconciliation control and financial workflow redesign (Dogbatsey & Ebhojie, 2018; Dogbatsey et al., 2019), financial analytics and data-driven executive decision systems (Lawal & Oduleye, 2018a; Lawal & Oduleye, 2019b), and tax governance and transfer pricing risk analytics (Lawal & Oduleye, 2018b; Lawal & Oduleye, 2019a) describes analytical workloads that scan large financial histories, which Section 3.6 identifies as among the most exposed to TDE overhead. Studies of web-based payment portal adoption (Agyei et al., 2019), multinational marketing law and consumer protection compliance (Anichukwueze et al., 2019), regulatory reform of statutory workflows (Uchechi et al., 2019), and banking product and relationship performance frameworks (Agboada, Ezeako, Okoli, & Ibeh, 2018; Ezeako et al., 2019) complete the picture of the regulated financial environment in which TDE is deployed.

Health information systems and data integrity

Healthcare is among the most heavily regulated contexts for encryption at rest. Evaluations of surveillance data quality on DHIS2, models for integrated laboratory information systems, and frameworks for traceability and chain-of-custody integrity in pharmaceutical distribution (Abdulrazaq et al., 2019; Aneke & Adesemoye, 2019; Ekemezie et al., 2018) describe data-intensive systems in which integrity, availability and confidentiality obligations coexist. Their reporting and analytical workloads typically involve wide scans of historical records, which makes them more sensitive to TDE overhead than conventional OLTP systems.

Infrastructure reliability, risk and vendor governance

Energy and utility operators increasingly fall within regulated estates, and their asset management and monitoring systems are database-backed. Studies of predictive maintenance and condition monitoring in critical power infrastructure, reliability engineering and failure analysis in complex systems, and risk-based process engineering for natural gas infrastructure (Adaramola et al., 2018; Asibor & Isong, 2019; Gideon et al., 2019) provide methods for weighing availability risk that apply equally to planning disruptive lifecycle events such as initial encryption. Data-driven safety risk control and predictive compliance monitoring (Adeyelu, 2018; Obriki & Arumosoye, 2018) illustrate the shift toward continuous, evidence-based oversight, while telecommunication network quality studies (Amayo & Popoola, 2017b) bear on the network dependency introduced by external key managers.

Finally, supply chain risk management and vendor governance frameworks (Agbabiaka et al., 2019; Okonkwo et al., 2019) are relevant because TDE is frequently a separately licensed option. In Oracle Database it forms part of the Advanced Security option, and in SQL Server it required Enterprise Edition until 2019, so vendor negotiation, licensing and supplier risk shape how widely organizations can afford to deploy it.

International literature on encrypted data management and storage performance

Research on encrypting data inside the database predates commercial TDE by more than two decades. Early designs for database encryption with subkeys (Davida et al., 1981) and later surveys of database security concepts and challenges (Bertino & Sandhu, 2005) established the trade-off that still governs TDE: the finer the granularity of encryption, the stronger the protection against some insiders, and the higher the processing cost. Work on balancing confidentiality and efficiency in untrusted relational systems (Damiani et al., 2003), on order-preserving encryption (Agrawal et al., 2004; Boldyreva et al., 2009; Popa et al., 2013), and on fast encryption of indexes in column-oriented systems (Ge & Zdonik, 2007) shows how much query functionality is lost or retained when encryption is applied above the page level. Searchable and homomorphic schemes (Curtmola et al., 2006; Gentry, 2009; Song et al., 2000) extend that functionality further but at costs that remain orders of magnitude above page-level encryption, which helps explain why TDE became the pragmatic choice for regulated estates.

A second strand evaluates encrypted query processing systems directly. Analytical queries over encrypted data (Tu et al., 2013), trusted-hardware databases (Arasu et al., 2015; Bajaj & Sion, 2011; Priebe et al., 2018) and architectures for concurrent, distributed access to encrypted cloud databases (Ferretti et al., 2014) report overheads that depend heavily on how much computation must occur on ciphertext, a cost TDE avoids by decrypting into the buffer cache. Practical guidance on implementing database encryption (Shmueli et al., 2014) reaches a similar conclusion about the importance of placement. At the same time, inference attacks on property-preserving encryption and analyses of why encrypted databases leak information (Grubbs et al., 2017; Naveed et al., 2015) are a reminder that TDE protects stored files and backups but not data served to authenticated sessions, the boundary emphasized in Section 3.1.

Storage and file-system encryption research provides the closest analogue to TDE outside the database engine. From the first cryptographic file system for UNIX (Blaze, 1993) through secure sharing on untrusted storage, enterprise-class encrypted file systems and cryptographic cloud storage (Halcrow, 2005; Kallahalla et al., 2003; Kamara & Lauter, 2010), these studies show that block-level encryption at the I/O boundary adds little latency once cipher operations are efficient. The relevant cipher and mode literature covers the design of AES (Daemen & Rijmen, 2002), authenticated and parallelizable modes such as OCB and GCM (McGrew & Viega, 2004; Rogaway et al., 2003), fast and timing-resistant AES-GCM implementations and the throughput gains of AES-NI (Akdemir et al., 2010; Käsper & Schwabe, 2009), and storage-specific modes and standards (Dworkin, 2010; IEEE, 2008). Together they explain why hardware acceleration is the dominant moderator identified in Section 3.6.

Key management and system context complete the picture. International guidance on designing cryptographic key management systems, on key management in cloud services and on security and

privacy controls (Barker et al., 2013; Chandramouli et al., 2013; Joint Task Force Transformation Initiative, 2013) frames the master key and data key hierarchy described in Section 3.2, while cold-boot attacks on encryption keys and cross-tenant information leakage in shared clouds (Halderman et al., 2009; Ristenpart et al., 2009) show why keys held in memory or on shared hosts need protection beyond the cipher itself. Classic work on database buffer management and write-ahead logging recovery (Effelsberg & Haerder, 1984; Mohan et al., 1992) explains why plaintext caching and redo generation shape TDE cost, and analyses of where OLTP engines spend their CPU time (Harizopoulos et al., 2008) show that cryptographic work is one small component among many. Virtualization and consolidation studies (Barham et al., 2003; Curino et al., 2011; Soror et al., 2008) bear on the masking of hardware features and on shared-resource effects, while deduplication research and work on deduplication-friendly encryption (Bellare et al., 2013; Douceur et al., 2002; Meyer & Bolosky, 2011; Zhu et al., 2008) explain why encrypted backups lose storage efficiency. Finally, benchmarking frameworks and guidance on fair database performance testing (Cooper et al., 2010; Raasveldt et al., 2018) inform the measurement protocol proposed in Section 3.12.2.

J. Conceptual model

Rationale

The synthesis shows that TDE overhead is not a fixed property of the cipher but an outcome of how the encryption mechanism interacts with workload, infrastructure and organizational practice. A model that makes those interactions explicit is needed for two reasons: it explains why published overhead figures diverge so widely, and it identifies which variables an organization must measure or control to predict its own outcome. This section therefore integrates the findings of Sections 3.4 to 3.9 into the TDE Performance Overhead Model (Figure 1).

Constructs

The model comprises seven constructs, each grounded in the synthesis.

- Regulatory drivers: the obligations and audit expectations that motivate encryption at rest, including PCI DSS, HIPAA, GDPR, NYDFS and NDPR (Section 3.4).
- Encryption scope and granularity: the share of the estate encrypted and the level at which encryption is applied (column, tablespace or database).
- TDE mechanism: page-level encryption at the I/O boundary with a plaintext buffer cache and a hierarchy of data encryption and master keys (Section 3.2).
- Moderators: hardware AES acceleration, buffer cache hit ratio, workload I/O intensity, CPU headroom, virtualization and interaction with compression and deduplication (Section 3.6).
- Steady-state overhead: the additional CPU consumption and response time incurred during normal operation (Section 3.5).
- Lifecycle overhead: the cost of initial encryption, backup and restore, key rotation, key manager dependency and shared-resource effects (Section 3.7).
- Governance and operational controls: key management policy, identity and access control, capacity planning, change scheduling and continuous audit and monitoring (Section 3.9).

© JUL 2019 | IRE Journals | Volume 3 Issue 1 | ISSN: 2456-8880
DOI: https://doi.org/10.64388/ire1725089

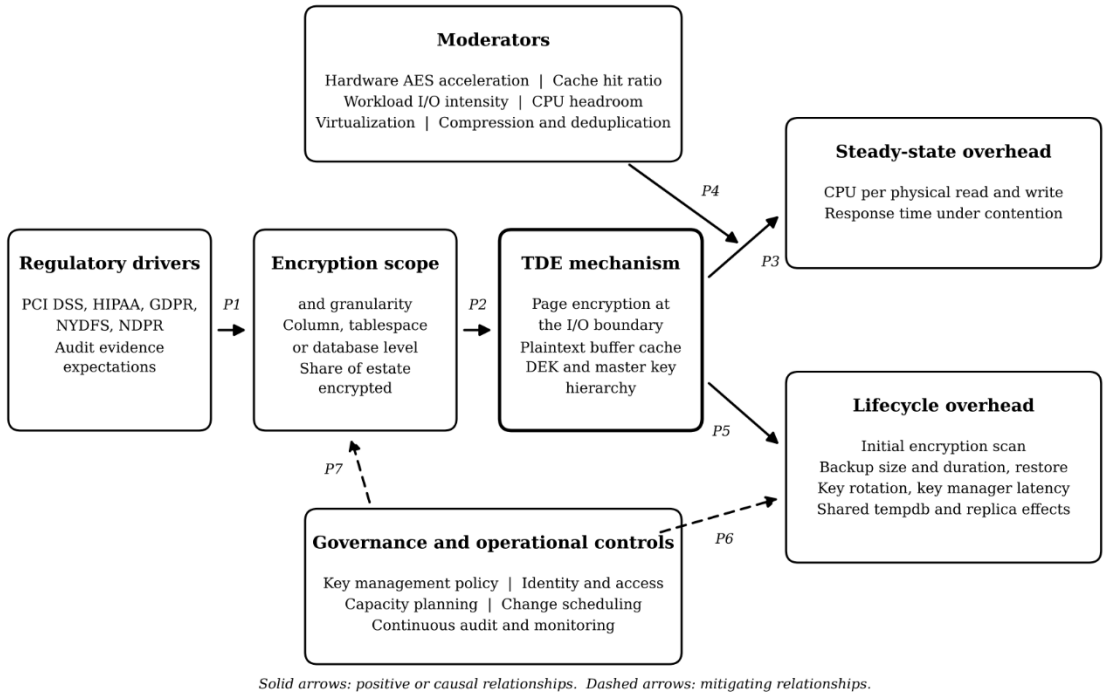


Figure 1. The TDE Performance Overhead Model

Propositions

The relationships in Figure 1 are stated as seven propositions (Table 8). They are offered as hypotheses for empirical testing rather than as established findings, because the evidence reviewed supports their direction more strongly than their magnitude.

Table 8. Propositions of the TDE Performance Overhead Model

No.	Proposition	Principal supporting evidence	Indicative measure
P1	Broader and stricter regulatory drivers increase the scope of encryption, favouring coarse-grained, estate-wide TDE that is easy to evidence.	Regulatory analysis (Table 5); Akomolafe & Agu, 2019a; Mbonu et al., 2018; Mbonu, Aliliele, et al., 2019	Share of databases and data volume encrypted
P2	Encryption scope determines aggregate cryptographic work: database or tablespace TDE lowers per-query cost relative to column encryption but raises the volume subject to lifecycle operations.	Oracle (2019); Shmueli et al. (2009); Elovici et al. (2004)	Encrypted data volume; query plan changes
P3	Because encryption occurs at the I/O boundary, steady-state overhead is proportional to physical rather than logical I/O.	Microsoft (2019a); Gueron (2010); Iyer et al. (2004)	CPU per physical page read and written
P4	Absence of hardware AES acceleration, low cache hit ratio, high I/O intensity, low CPU headroom and virtualization that masks AES-NI amplify steady-state overhead.	Table 6; Gueron (2010); Ahmed & Odejobi, 2018a; Ahmed & Odejobi, 2018b; Odejobi & Ahmed, 2018b	Overhead difference across hardware and cache configurations

No.	Proposition	Principal supporting evidence	Indicative measure
P5	For large estates, lifecycle overhead exceeds steady-state overhead as a source of performance and availability risk.	Table 7; Microsoft (2019a, 2019b); Oracle (2019)	Initial encryption, backup and restore duration
P6	Governance and operational controls, including master key rotation policy, change scheduling and capacity planning, mitigate lifecycle overhead.	Barker (2016); Agboada et al., 2019; Anyanwu, 2019; Dosunmu & Ogundele, 2019	Unplanned outage minutes and SLA breaches during rollout
P7	Risk-based governance moderates the scope decision, reducing encryption that exceeds regulatory need.	Akomolafe & Agu, 2018; Akomolafe & Agu, 2019a; Fadayomi et al., 2019	Encrypted scope relative to classified sensitive data

Taken together, the propositions imply that two organizations running the same database version with the same cipher can experience very different performance outcomes. The difference is explained by the scope they choose to encrypt, the hardware and workload characteristics that moderate the mechanism, and the governance practices with which they manage lifecycle events.

K. Discussion

Reconciling divergent reports

The review helps reconcile the apparent contradiction between vendor estimates of low overhead and practitioner reports of significant impact. The two are largely measuring different things. Vendor estimates describe steady-state, cache-friendly operation on accelerated hardware. Practitioner concerns usually arise from lifecycle events (initial encryption, backup windows, restores without keys), from I/O-heavy workloads with low cache residency, from consolidated instances where shared resources such as tempdb become encrypted, and from older or virtualized hosts without exposed AES acceleration. Framing the cost of TDE as a single percentage therefore obscures more than it reveals. A more useful framing separates a usually small steady-state CPU tax from a set of potentially large but schedulable and mitigable lifecycle costs.

TDE overhead as a system property

The conceptual model reframes TDE performance as a property of the system in which it is deployed rather than of the encryption algorithm. On accelerated hardware the cryptographic work itself is small (P3), so the variables that matter most are those the model places around the mechanism: the scope selected in response to regulation (P1, P2), the moderators that determine how much physical I/O must be encrypted and how readily the processor absorbs it (P4), and the lifecycle events whose cost depends on estate size and operational planning (P5, P6). This explains why the same product can be described as nearly free by its vendor and as disruptive by practitioners who encountered it during rollout.

The model also clarifies the role of scoping. Broad, database-level encryption simplifies audit evidence and avoids the query-processing penalties of column-level encryption, but it increases the volume of data subject to lifecycle operations. Risk-based scoping (P7), supported by the audit and internal control literature (Akomolafe & Agu, 2018; Akomolafe & Agu, 2019a), offers a way to balance these pressures without weakening compliance.

Relationship with the adjacent literature

The adjacent-domain literature reinforces the view that encryption at rest is one control within a wider assurance system. Risk-based assurance and security audit frameworks emphasize availability alongside confidentiality (Dosunmu & Ogundele, 2019; Fadayomi et al., 2019), identity and directory studies locate the residual risk in access to keys rather than in the cipher (Ladapo et al., 2018; Oshoba et al., 2019), and IT operations and platform governance research describes the change disciplines on which lifecycle mitigation depends (Agboada et al., 2019; Anyanwu, 2019; Badmus et al., 2019b). None of these studies measures TDE directly, which underlines the need for the empirical work identified in Section 3.14.

L. Implications

Implications for practitioners

Three implications follow for organizations deploying TDE under regulatory obligation. First, performance risk is concentrated in rollout and in operations rather than in day-to-day query processing, so programme plans should allocate engineering effort accordingly: to scheduling initial encryption, revalidating backup and recovery objectives and hardening key management. Second, hardware refresh and virtualization configuration are part of the encryption control, because the presence of hardware

AES acceleration largely determines whether vendor estimates hold. Third, compliance teams and infrastructure teams should agree an interpretation of key rotation requirements that distinguishes master key rotation from data key rotation, avoiding unnecessary recurring re-encryption. The security audit, risk-based assurance and reliability engineering literature offers practical means of making these trade-offs explicit to boards and auditors (Dosunmu & Ogundele, 2019; Fadayomi et al., 2019; Gideon et al., 2019).

These implications translate into the following deployment recommendations.

- Prefer tablespace or database-level TDE over column-level encryption where regulatory scope permits, to minimize query-processing overhead.
- Size buffer caches to keep hot working sets resident, which reduces both baseline I/O and encryption cost.
- Plan initial encryption as a change event: use suspend and resume capabilities or online conversion where available, encrypt standbys before primaries where architectures allow, and monitor log generation.
- Revalidate backup windows and storage capacity; enable engine-level compression compatible with TDE and adjust deduplication assumptions for backup appliances.
- Avoid co-locating encrypted and latency-sensitive unencrypted databases on instances where encryption of shared resources (for example, SQL Server tempdb) would affect the latter.
- Adopt a key hierarchy policy that rotates master keys on schedule and reserves data key rotation for suspected compromise, documented in agreement with auditors and aligned with NIST SP 800-57 cryptoperiod guidance.
- Treat the key manager or HSM as tier-one infrastructure with high availability, monitoring and rehearsed recovery, since loss of keys renders data and backups unrecoverable.
- Forward security telemetry on key access, key retrieval failures and encryption state changes to the security operations centre.

The last two recommendations draw on security operations and identity governance research (Ladapo et al., 2019; Mbonu, Iwuanyanwu, et al., 2019; Oshoba et al., 2019).

Measurement protocol for practitioners

Given the weakness of independent evidence, regulated organizations should measure TDE overhead in their own environments before estate-wide rollout. The following framework, derived from the review findings, is proposed.

- Confirm hardware AES acceleration is present and exposed to the database host, including inside virtual machines.
- Select representative workloads: replay of captured production OLTP traffic, the heaviest batch and reporting jobs, and a standardized benchmark such as TPC-C or TPC-H for comparability.
- Establish an unencrypted baseline on identical hardware and data, and repeat each run several times to estimate variance.
- Record CPU utilization, throughput, 95th and 99th percentile response time, physical reads and writes, and buffer cache hit ratio.
- Measure lifecycle operations separately: elapsed time and resource consumption of initial encryption, full and incremental backup size and duration, restore duration, and database startup time with the chosen key manager.
- Report results as a steady-state overhead and a lifecycle cost profile rather than as a single percentage.
- Integrate the protocol into existing continuous control validation and CI/CD practice so that measurements are repeated after platform upgrades.

This approach aligns with security audit and automated monitoring practice (Dosunmu & Ogundele, 2019; Mbonu, Iwuanyanwu, et al., 2019) and with DevOps governance practices for regulated platforms (Badmus et al., 2018; Badmus et al., 2019a).

Implications for auditors and regulators

For auditors, the review suggests that evidence of encryption should extend beyond the fact that TDE is enabled. Auditors can reasonably ask whether key management separates master key and data key rotation, whether keys are escrowed and restores rehearsed, and whether recovery time objectives were revalidated after encryption was introduced. For regulators and standards bodies, the finding that master key rotation achieves most of the security benefit of rotation at a fraction of the performance cost suggests that guidance on cryptoperiods could usefully distinguish between key types, as NIST guidance already does (Barker, 2016). Clearer statements on whether file-level encryption satisfies particular requirements would also reduce the tendency to encrypt more broadly than risk requires.

Implications for theory and research

For researchers, the review contributes a taxonomy and a set of propositions that move the question from "how much does TDE cost" to "under which

conditions, and in which phases of the lifecycle, does TDE cost what". The model positions encryption overhead at the intersection of database systems research, which explains the mechanism, and information systems governance research, which explains scope and lifecycle management. Treating these literatures together offers a basis for studies that combine controlled benchmarking with organizational case research.

M. Limitations

Several limitations apply to this review. Selection bias may arise from the inclusion of vendor documentation, which has an interest in presenting favourable performance; this was mitigated by explicit quality scoring on independence and by triangulation against first-principles analysis and academic work. Publication bias may suppress unfavourable findings, particularly given historical benchmark publication restrictions. Construct validity is threatened by inconsistent definitions of overhead across sources, which variously report CPU utilization, throughput, response time or elapsed time. Finally, the rapid evolution of products means that findings documented in 2019 may not apply to earlier versions still running in many estates, or to later releases. In addition, the regional seed corpus was screened by title from a single consolidated list and is weighted toward conceptual frameworks and reviews, while the international seed corpus was assembled purposively rather than by exhaustive search, so both should be read as contextual rather than exhaustive.

The conceptual model carries limitations of its own. Its propositions are derived inductively from a heterogeneous and largely vendor-authored evidence base, so their direction is better supported than their magnitude, and some constructs, notably governance and operational controls, aggregate practices that may behave differently when measured separately. Screening, extraction and quality scoring were performed by a single reviewer, which may introduce judgement bias despite the use of explicit criteria. Finally, the review covers publications up to mid-2019, so developments after that date fall outside its scope.

N. Future research

The review identifies a clear research agenda: independent, reproducible measurement of commercial TDE implementations on standardized OLTP and analytical benchmarks; explicit reporting of hardware acceleration, cache sizing and variance; measurement of lifecycle operations such as initial encryption and backup at realistic scale; and evaluation of virtualized and cloud-managed database services, where encryption at rest is increasingly enabled by default and the customer has limited visibility into its implementation.

The propositions in Table 8 suggest specific lines of inquiry:

- Controlled benchmarks on TPC-C and TPC-H style workloads with and without hardware AES acceleration and with varied buffer cache sizing, to estimate the magnitude of P3 and P4.
- Field measurement of initial encryption, backup and restore duration in production-scale estates, to test P5 and quantify lifecycle cost relative to steady-state cost.
- Comparative case studies of organizations with different key rotation and change-scheduling practices, to test the mitigating effect of governance proposed in P6.
- Survey or archival research linking regulatory regime and audit practice to encryption scope decisions, to test P1 and P7.
- Evaluation of encryption at rest in virtualized and cloud-managed database services, where hardware exposure and key custody differ from on-premises deployments.

IV. CONCLUSION

Transparent Data Encryption is the most widely deployed technical control for encryption at rest in regulated enterprise database estates, valued for its ability to satisfy auditors without application change. This systematic review finds that its steady-state performance cost is, on modern hardware with AES acceleration and well-cached workloads, consistently characterized as small, because encryption is applied at page granularity at the I/O boundary. The more significant costs lie in lifecycle events, in I/O-intensive workloads, in shared-resource side effects and in key management dependencies, all of which are predictable and largely mitigable through planning. The review also finds that the evidence base remains dominated by vendor documentation, with a notable lack of independent, reproducible measurement. Until that gap is closed, regulated organizations are best served by treating published overhead figures as indicative, by measuring in their own environments using a protocol that separates steady-state from lifecycle costs, and by recognizing that the infrastructure and key management decisions surrounding TDE matter as much to its performance as the cipher itself. The TDE Performance Overhead Model and its seven propositions offer a basis for testing these conclusions empirically and for structuring deployment decisions in the meantime.

REFERENCES

[1] Abdulrazaq, K., Yusuf, H. D., Osinem, H. E., & Eribenne, F. (2019). A model for improving infection detection accuracy through integrated laboratory information systems. Iconic Research and Engineering Journals, 3(6), 650-675. https://doi.org/10.64388/IREV3I6-1723089

[2] Adaramola, T. S., Fadero, S., & Gideon, E. N. (2018). Predictive maintenance and condition monitoring in critical power and energy infrastructure. Iconic Research and Engineering Journals, 2(5), 454-477. https://doi.org/10.64388/IREV2I5-1722489

[3] Adeyelu, O. O. (2018). A predictive compliance monitoring framework for detecting systemic safety risks through aviation consumer complaint data. Iconic Research and Engineering Journals, 1(8), 250-276. https://doi.org/10.64388/IREV1I8-1718478

[4] Agbabiaka, J., Okonkwo, C. S., Ogunwole, O., Mayo, W., & Okeke, O. T. (2019). Supply chain risk management model for EPC and gas processing projects. Iconic Research and Engineering Journals, 3(2), 968-980. https://doi.org/10.64388/IREV3I2-1713124

[5] Agboada, P., Ezeako, R., Okoli, I. E., & Ibeh, A. V. (2018). Distributorship and trade finance product design: A conceptual framework for new-to-bank growth in emerging markets. Iconic Research and Engineering Journals, 2(5), 552-578. https://doi.org/10.64388/IREV2I5-1723312

[6] Agboada, P., Ezeako, R., Okoli, I. E., & Uzondu, C. S. (2018). Digital transformation in operations management: A review of advances in automation, analytics, and process redesign. International Journal of Engineering Technology Research & Management, 2(11), 109-134.

[7] Agboada, P., Ezeako, R., Okoli, I. E., & Uzondu, C. S. (2019). Standard operating procedures as change infrastructure: A conceptual framework for sustaining operational improvement. Iconic Research and Engineering Journals, 3(6), 706-730. https://doi.org/10.64388/IREV3I6-1723313

[8] Agrawal, R., Kiernan, J., Srikant, R., & Xu, Y. (2004). Order preserving encryption for numeric data. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data* (pp. 563-574). ACM.

[9] Agyei, R. K., Izuchukwu, O. T., Feboh, D., & Idowu, A. (2019). Determinants of web-based payment portal adoption in Sub-Saharan Africa: A review and synthesis. International Journal of Computer Science and Mathematical Theory, 5(2), 56-79. https://doi.org/10.56201/ijcsmt.vol.5.no2.2019.pg56.79

[10] Ahmed, K. S., & Odejobi, O. D. (2018a). Conceptual framework for scalable and secure cloud architectures for enterprise messaging. IRE Journals, 2(1), 1-15.

[11] Ahmed, K. S., & Odejobi, O. D. (2018b). Resource allocation model for energy-efficient virtual machine placement in data centers. IRE Journals, 2(3), 1-10.

[12] Ahmed, K. S., Odejobi, O. D., & Oshoba, T. O. (2019). Algorithmic model for constraint satisfaction in cloud network resource allocation. IRE Journals, 2(12), 516-532.

[13] Akdemir, K., Dixon, M., Feghali, W., Fay, P., Gopal, V., Guilford, J., Ozturk, E., Wolrich, G., & Zohar, R. (2010). *Breakthrough AES performance with Intel AES New Instructions* [White paper]. Intel Corporation.

[14] Akeju, B., Edivri, J., Ogbole, J. I., Okoruwa, P. O., Fadayomi, O., & Abolaji, T. O. (2018). Conceptual model for insider threat classification and risk modeling in complex digital systems. Iconic Research and Engineering Journals, 1(9), 476-492. https://doi.org/10.64388/IREV1I9-1713778

[15] Akomolafe, O., & Agu, M. U. (2018). A conceptual model for enhancing internal audit quality through technology-enabled risk assessment frameworks. Iconic Research and Engineering Journals, 1(9), 458-475.

[16] Akomolafe, O., & Agu, M. U. (2019a). A conceptual framework for developing risk-based internal control models in the insurance and banking sectors. Iconic Research and Engineering Journals, 2(8), 335-354.

[17] Akomolafe, O., & Agu, M. U. (2019b). A review of data-driven risk evaluation models for emerging market financial institutions. Iconic Research and Engineering Journals, 3(6), 433-448.

[18] Akomolafe, O., & Agu, M. U. (2019c). Advances in financial resilience through integrated governance and compliance strategies. Iconic Research and Engineering Journals, 2(10), 607-620.

[19] Amayo, E. B., & Popoola, T. T. (2017b). Scientific study of telecommunication deployment in achieving quality network.

International Journal of Trend in Research and Development, 4(5), 311-314.

[20] Aneke, O. B., & Adesemoye, A. C. (2019). Toward an integrated conceptual framework for traceability and chain-of-custody integrity in pharmaceutical distribution. Iconic Research and Engineering Journals, 3(4), 655-672.

[21] Anichukwueze, C. C., Osuji, V. C., & Oguntegbe, E. E. (2019). Global marketing law and consumer protection challenges: A strategic framework for multinational compliance. IRE Journals, 3(6), 325-333.

[22] Anyanwu, K. C. E. (2019). Developing a maturity model for standardizing IT support operations across geographically distributed multi-office organizations. Iconic Research and Engineering Journals, 3(5)., 580-598.

[23] Arasu, A., Blanas, S., Eguro, K., Kaushik, R., Kossmann, D., Ramamurthy, R., & Venkatesan, R. (2013). Orthogonal security with Cipherbase. In *Proceedings of the 6th Biennial Conference on Innovative Data Systems Research (CIDR)*.

[24] Arasu, A., Eguro, K., Joglekar, M., Kaushik, R., Kossmann, D., & Ramamurthy, R. (2015). Transaction processing on confidential data using Cipherbase. In *Proceedings of the 31st IEEE International Conference on Data Engineering (ICDE)*. IEEE.

[25] Asibor, E., & Isong, D. E. (2019). Risk-based process engineering framework for strengthening reliability across complex natural gas infrastructure systems. Iconic Research and Engineering Journals, 3(2), 1020-1060. https://doi.org/10.64388/IREV3I2-1723153

[26] Badmus, O., Dosunmu, A. A., & Ozowara, D. E. (2018). A systematic review of CI/CD pipeline strategies in Salesforce DevOps: Tools, practices, and deployment outcomes. Iconic Research and Engineering Journals, 2(6).

[27] Badmus, O., Dosunmu, A. A., & Ozowara, D. E. (2019a). A conceptual model for ETL design and data integration in Salesforce-centric enterprise architectures. Iconic Research and Engineering Journals, 3(5).

[28] Badmus, O., Dosunmu, A. A., & Ozowara, D. E. (2019b). A governance framework for Salesforce platform management in regulated healthcare environments. Iconic Research and Engineering Journals, 3(6).

[29] Bajaj, S., & Sion, R. (2011). TrustedDB: A trusted hardware based database with privacy and data confidentiality. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*. ACM.

[30] Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., & Warfield, A. (2003). Xen and the art of virtualization. In *Proceedings of the 19th ACM Symposium on Operating Systems Principles* (pp. 164-177). ACM.

[31] Barker, E. (2016). *Recommendation for key management, Part 1: General* (NIST Special Publication 800-57 Part 1, Revision 4). National Institute of Standards and Technology.

[32] Barker, E., Smid, M., Branstad, D., & Chokhani, S. (2013). *A framework for designing cryptographic key management systems* (NIST Special Publication 800-130). National Institute of Standards and Technology.

[33] Bellare, M., Keelveedhi, S., & Ristenpart, T. (2013). Message-locked encryption and secure deduplication. In *Advances in Cryptology: EUROCRYPT 2013, Lecture Notes in Computer Science* (Vol. 7881). Springer.

[34] Bertino, E., & Sandhu, R. (2005). Database security: Concepts, approaches, and challenges. *IEEE Transactions on Dependable and Secure Computing, 2*(1), 2-19.

[35] Blaze, M. (1993). A cryptographic file system for UNIX. In *Proceedings of the 1st ACM Conference on Computer and Communications Security* (pp. 9-16). ACM.

[36] Bola-Sadipe, D., Aliliele, C., & Eboh, E. E. (2019). Conceptual model for strategic accounting systems strengthening financial transparency and regulatory compliance. IRE Journals, 3(6), 535-564. https://doi.org/10.64388/IREV3I6-1715327

[37] Boldyreva, A., Chenette, N., Lee, Y., & O'Neill, A. (2009). Order-preserving symmetric encryption. In *Advances in Cryptology: EUROCRYPT 2009, Lecture Notes in Computer Science* (Vol. 5479, pp. 224-241). Springer.

[38] Bouganim, L., & Pucheral, P. (2002). Chip-secured data access: Confidential data on untrusted servers. In *Proceedings of the 28th International Conference on Very Large Data Bases (VLDB)* (pp. 131-142).

[39] Chandramouli, R., Iorga, M., & Chokhani, S. (2013). *Cryptographic key management issues and challenges in cloud services* (NIST

Interagency Report 7956). National Institute of Standards and Technology.

[40] Cooper, B. F., Silberstein, A., Tam, E., Ramakrishnan, R., & Sears, R. (2010). Benchmarking cloud serving systems with YCSB. In *Proceedings of the 1st ACM Symposium on Cloud Computing* (pp. 143-154). ACM.

[41] Curino, C., Jones, E. P. C., Madden, S., & Balakrishnan, H. (2011). Workload-aware database monitoring and consolidation. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data* (pp. 313-324). ACM.

[42] Curtmola, R., Garay, J., Kamara, S., & Ostrovsky, R. (2006). Searchable symmetric encryption: Improved definitions and efficient constructions. In *Proceedings of the 13th ACM Conference on Computer and Communications Security* (pp. 79-88). ACM.

[43] Daemen, J., & Rijmen, V. (2002). *The design of Rijndael: AES, the Advanced Encryption Standard*. Springer.

[44] Damiani, E., De Capitani di Vimercati, S., Jajodia, S., Paraboschi, S., & Samarati, P. (2003). Balancing confidentiality and efficiency in untrusted relational DBMSs. In *Proceedings of the 10th ACM Conference on Computer and Communications Security* (pp. 93-102). ACM.

[45] Davida, G. I., Wells, D. L., & Kam, J. B. (1981). A database encryption system with subkeys. *ACM Transactions on Database Systems, 6*(2), 312-328.

[46] Dogbatsey, E. A., & Ebhojie, O. (2018). Budget compliance and statutory reporting in Sub-Saharan Africa: A systematic review of frameworks, gaps, and reform pathways. Zenodo. https://doi.org/10.5281/zenodo.20446859

[47] Dogbatsey, E. A., Ebhojie, O., & Oyeleye, A. O. (2019). Reconciliation control and financial workflow redesign in emerging market organizations: A conceptual framework. Zenodo. https://doi.org/10.5281/zenodo.20447103

[48] Dosunmu, A. A., & Ogundele, P. O. (2019). Security audit and enterprise risk assessment frameworks for resilient information systems. Iconic Research and Engineering Journals, 3(5), 434-447. https://doi.org/10.64388/IREV3I5-1713225

[49] Douceur, J. R., Adya, A., Bolosky, W. J., Simon, D., & Theimer, M. (2002). Reclaiming space from duplicate files in a serverless distributed file system. In *Proceedings of the 22nd International Conference on Distributed Computing Systems* (pp. 617-624). IEEE.

[50] Dworkin, M. (2001). *Recommendation for block cipher modes of operation: Methods and techniques* (NIST Special Publication 800-38A). National Institute of Standards and Technology.

[51] Dworkin, M. (2010). *Recommendation for block cipher modes of operation: The XTS-AES mode for confidentiality on storage devices* (NIST Special Publication 800-38E). National Institute of Standards and Technology.

[52] Effelsberg, W., & Haerder, T. (1984). Principles of database buffer management. *ACM Transactions on Database Systems, 9*(4), 560-595.

[53] Ekemezie, U. C., Ohunyon, G., & Sydney, E. (2018). Evaluating IDSR implementation in Benue State, Nigeria: DHIS2 data quality and surveillance capacity. Iconic Research and Engineering Journals, 2(6).

[54] Elovici, Y., Waisenberg, R., Shmueli, E., & Gudes, E. (2004). A structure preserving database encryption scheme. In *Secure Data Management (SDM 2004), Lecture Notes in Computer Science* (Vol. 3178, pp. 28-40). Springer.

[55] European Parliament and Council of the European Union. (2016). Regulation (EU) 2016/679 (General Data Protection Regulation). *Official Journal of the European Union*, L 119.

[56] Ezeako, R., Okoli, I. E., Uzondu, C. S., & Ibeh, A. V. (2019). Relationship banking performance: A conceptual framework for portfolio growth and profitability acceleration. International Journal of Engineering Technology Research & Management, 3(11), 132-151.

[57] Fadayomi, O., Abolaji, T. O., Edivri, J., Ogbole, J. I., Okoruwa, P. O., & Akeju, B. (2019). Risk-based cybersecurity assurance and data availability: Limitations, advances and future research opportunities. Iconic Research and Engineering Journals, 2(12), 602-617. https://doi.org/10.64388/IREV2I12-1713779

[58] Ferretti, L., Colajanni, M., & Marchetti, M. (2014). Distributed, concurrent, and independent access to encrypted cloud databases. *IEEE Transactions on Parallel and Distributed Systems, 25*(2), 437-446.

[59] Garousi, V., Felderer, M., & Mäntylä, M. V. (2019). Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. *Information and Software Technology, 106*, 101-121.

[60] Ge, T., & Zdonik, S. (2007). Fast, secure encryption for indexing in a column-oriented DBMS. In *Proceedings of the 23rd IEEE International Conference on Data Engineering (ICDE)*. IEEE.

[61] Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing* (pp. 169-178). ACM.

[62] Gideon, E. N., Adaramola, T. S., & Fadero, S. (2019). Reliability engineering and failure analysis in complex engineering systems. Iconic Research and Engineering Journals, 2(11), 703-727. https://doi.org/10.64388/IREV2I11-1722490

[63] Grubbs, P., Ristenpart, T., & Shmatikov, V. (2017). Why your encrypted database is not secure. In *Proceedings of the 16th Workshop on Hot Topics in Operating Systems (HotOS)* (pp. 162-168). ACM.

[64] Gueron, S. (2010). *Intel Advanced Encryption Standard (AES) New Instructions Set* (White paper, Revision 3.0). Intel Corporation.

[65] Hacigumus, H., Iyer, B., Li, C., & Mehrotra, S. (2002). Executing SQL over encrypted data in the database-service-provider model. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data* (pp. 216-227).

[66] Halcrow, M. A. (2005). eCryptfs: An enterprise-class encrypted filesystem for Linux. In *Proceedings of the Linux Symposium* (Vol. 1, pp. 209-226).

[67] Halderman, J. A., Schoen, S. D., Heninger, N., Clarkson, W., Paul, W., Calandrino, J. A., Feldman, A. J., Appelbaum, J., & Felten, E. W. (2009). Lest we remember: Cold-boot attacks on encryption keys. *Communications of the ACM, 52*(5), 91-98.

[68] Harizopoulos, S., Abadi, D. J., Madden, S., & Stonebraker, M. (2008). OLTP through the looking glass, and what we found there. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data* (pp. 981-992). ACM.

[69] IBM. (2019). *Db2 native encryption*. IBM Db2 11.5 Knowledge Center.

[70] IEEE. (2008). *IEEE standard for cryptographic protection of data on block-oriented storage devices* (IEEE Std 1619-2007). Institute of Electrical and Electronics Engineers.

[71] Iyer, B., Mehrotra, S., Mykletun, E., Tsudik, G., & Wu, Y. (2004). A framework for efficient storage security in RDBMS. In *Advances in Database Technology (EDBT 2004), Lecture Notes in Computer Science* (Vol. 2992, pp. 147-164). Springer.

[72] Joint Task Force Transformation Initiative. (2013). *Security and privacy controls for federal information systems and organizations* (NIST Special Publication 800-53, Revision 4). National Institute of Standards and Technology.

[73] Kallahalla, M., Riedel, E., Swaminathan, R., Wang, Q., & Fu, K. (2003). Plutus: Scalable secure file sharing on untrusted storage. In *Proceedings of the 2nd USENIX Conference on File and Storage Technologies (FAST)* (pp. 29-42). USENIX Association.

[74] Kamara, S., & Lauter, K. (2010). Cryptographic cloud storage. In *Financial Cryptography and Data Security: FC 2010 Workshops, Lecture Notes in Computer Science* (Vol. 6054, pp. 136-149). Springer.

[75] Käsper, E., & Schwabe, P. (2009). Faster and timing-attack resistant AES-GCM. In *Cryptographic Hardware and Embedded Systems: CHES 2009, Lecture Notes in Computer Science* (Vol. 5747, pp. 1-17). Springer.

[76] Kitchenham, B., & Charters, S. (2007). *Guidelines for performing systematic literature reviews in software engineering* (EBSE Technical Report EBSE-2007-01). Keele University and Durham University.

[77] Kitchenham, B., Brereton, O. P., Budgen, D., Turner, M., Bailey, J., & Linkman, S. (2009). Systematic literature reviews in software engineering: A systematic literature review. *Information and Software Technology, 51*(1), 7-15.

[78] Ladapo, O. O., Dosunmu, A. A., Jooda, D., & Abolaji, T. O. (2018). Lessons learned from offline assessment of security-critical systems: The case of Microsoft Active Directory. Iconic Research and Engineering Journals, 2(6), 277-299. https://doi.org/10.64388/IREV2I6-1717205

[79] Ladapo, O. O., Jooda, D., Dosunmu, A. A., & Abolaji, T. O. (2019). Implementation of

Active Directory for efficient management of enterprise networks. Iconic Research and Engineering Journals, 3(4), 608-627. https://doi.org/10.64388/IREV3I4-1717206

[80] Lawal, O. A., & Oduleye, T. E. (2018a). A conceptual model for financial analytics driven enterprise value creation in technology firms. Iconic Research and Engineering Journals, 2(2).

[81] Lawal, O. A., & Oduleye, T. E. (2018b). A review and conceptual framework for tax governance and cross-border compliance analytics. Iconic Research and Engineering Journals, 2(5).

[82] Lawal, O. A., & Oduleye, T. E. (2019a). A conceptual risk assessment model for transfer pricing in multinational corporations. Iconic Research and Engineering Journals, 2(12).

[83] Lawal, O. A., & Oduleye, T. E. (2019b). Conceptualizing data driven executive decision systems for strategic financial planning. Iconic Research and Engineering Journals, 3(3).

[84] Mbonu, I. S., Aliliele, C., Iwuanyanwu, U., & Oluoha, O. M. (2018). A conceptual framework for legal and ethical risk modeling in enterprise data protection governance systems. Iconic Research and Engineering Journals, 2(2), 207-226. https://doi.org/10.64388/IREV2I2-1714911

[85] Mbonu, I. S., Aliliele, C., Uzoka, E., & Oluoha, O. M. (2019). A review of comparative data protection regulations and secure cloud implementation strategies across jurisdictions. Iconic Research and Engineering Journals, 2(9), 482-501. https://doi.org/10.64388/IREV2I9-1714912

[86] Mbonu, I. S., Iwuanyanwu, U., Uzoka, E., & Oluoha, O. M. (2019). Advances in enterprise log analytics and automated incident response architectures using Python and SIEM platforms. Iconic Research and Engineering Journals, 3(2), 1000-1019. https://doi.org/10.64388/IREV3I2-1714915

[87] McGrew, D. A., & Viega, J. (2004). The security and performance of the Galois/Counter Mode (GCM) of operation. In *Progress in Cryptology: INDOCRYPT 2004, Lecture Notes in Computer Science* (Vol. 3348, pp. 343-355). Springer.

[88] Mensah, K. O., Pedro, O., Kinyanjui, A., Okoli, I., & Agboada, P. (2019). A review of capital adequacy frameworks and risk weighted asset optimization in modern banking systems. Iconic Research and Engineering Journals, 3(6), 731-768. https://doi.org/10.64388/IREV3I6-1723267

[89] Meyer, D. T., & Bolosky, W. J. (2011). A study of practical deduplication. In *Proceedings of the 9th USENIX Conference on File and Storage Technologies (FAST)*. USENIX Association.

[90] Microsoft. (2019a). *Transparent data encryption (TDE)*. SQL Server documentation, Microsoft Docs.

[91] Microsoft. (2019b). *Backup compression (SQL Server)*. SQL Server documentation, Microsoft Docs.

[92] Mohan, C., Haderle, D., Lindsay, B., Pirahesh, H., & Schwarz, P. (1992). ARIES: A transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging. *ACM Transactions on Database Systems, 17*(1), 94-162.

[93] Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., & The PRISMA Group. (2009). Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement. *PLoS Medicine, 6*(7), e1000097.

[94] MongoDB, Inc. (2019). *Encryption at rest*. MongoDB Manual, version 4.2.

[95] National Information Technology Development Agency. (2019). *Nigeria Data Protection Regulation*. NITDA.

[96] National Institute of Standards and Technology. (2001). *Advanced Encryption Standard (AES)* (FIPS Publication 197).

[97] Naveed, M., Kamara, S., & Wright, C. V. (2015). Inference attacks on property-preserving encrypted databases. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security* (pp. 644-655). ACM.

[98] New York State Department of Financial Services. (2017). *Cybersecurity requirements for financial services companies* (23 NYCRR Part 500).

[99] Obriki, O. D., & Arumosoye, O. M. (2018). Conceptual modeling of data-driven occupational safety risk control in large-scale energy infrastructure projects. Iconic Research and Engineering Journals, 1(7), 169-189. https://doi.org/10.64388/IREV1I7-1714414

[100] Odejobi, O. D., & Ahmed, K. S. (2018b). Performance evaluation model for multi-tenant Microsoft 365 deployments under high concurrency. IRE Journals, 1(11), 92-107.

[101] Odejobi, O. D., Hammed, N. I., & Ahmed, K. S. (2019). Approximation complexity model for cloud-based database optimization problems. IRE Journals, 2(9), 1-10.

[102] Okonkwo, C. S., Ogunwole, O., Okeke, O. T., & Mayo, W. (2019). Conceptual framework for cost reduction through contract negotiation and vendor governance. Iconic Research and Engineering Journals, 2(9), 468-482. https://doi.org/10.64388/IREV2I9-1713121

[103] Oracle Corporation. (2012). *Oracle Advanced Security transparent data encryption best practices* [White paper].

[104] Oracle Corporation. (2019). *MySQL 8.0 reference manual: InnoDB data-at-rest encryption*.

[105] Oracle Corporation. (2019). *Oracle Database Advanced Security guide, 19c*. Oracle Help Center.

[106] Oshoba, T. O., Hammed, N. I., & Odejobi, O. D. (2019). Secure identity and access management model for distributed and federated systems. IRE Journals, 3(4), 550-567.

[107] PCI Security Standards Council. (2018). *Payment Card Industry Data Security Standard: Requirements and security assessment procedures, version 3.2.1*.

[108] Popa, R. A., Li, F. H., & Zeldovich, N. (2013). An ideal-security protocol for order-preserving encoding. In *Proceedings of the 2013 IEEE Symposium on Security and Privacy* (pp. 463-477). IEEE.

[109] Popa, R. A., Redfield, C. M. S., Zeldovich, N., & Balakrishnan, H. (2011). CryptDB: Protecting confidentiality with encrypted query processing. In *Proceedings of the 23rd ACM Symposium on Operating Systems Principles (SOSP)* (pp. 85-100).

[110] Priebe, C., Vaswani, K., & Costa, M. (2018). EnclaveDB: A secure database using SGX. In *Proceedings of the 2018 IEEE Symposium on Security and Privacy* (pp. 264-278). IEEE.

[111] Raasveldt, M., Holanda, P., Gubner, T., & Mühleisen, H. (2018). Fair benchmarking considered difficult: Common pitfalls in database performance testing. In *Proceedings of the Workshop on Testing Database Systems (DBTest)*, Article 2. ACM.

[112] Ristenpart, T., Tromer, E., Shacham, H., & Savage, S. (2009). Hey, you, get off of my cloud: Exploring information leakage in third-party compute clouds. In *Proceedings of the 16th ACM Conference on Computer and Communications Security* (pp. 199-212). ACM.

[113] Rogaway, P., Bellare, M., & Black, J. (2003). OCB: A block-cipher mode of operation for efficient authenticated encryption. *ACM Transactions on Information and System Security, 6*(3), 365-403.

[114] Scarfone, K., Souppaya, M., & Sexton, M. (2007). *Guide to storage encryption technologies for end user devices* (NIST Special Publication 800-111). National Institute of Standards and Technology.

[115] Shmueli, E., Vaisenberg, R., Elovici, Y., & Glezer, C. (2009). Database encryption: An overview of contemporary challenges and design considerations. *ACM SIGMOD Record, 38*(3), 29-34.

[116] Shmueli, E., Vaisenberg, R., Gudes, E., & Elovici, Y. (2014). Implementing a database encryption solution, design and implementation issues. *Computers & Security, 44*, 33-50.

[117] Song, D. X., Wagner, D., & Perrig, A. (2000). Practical techniques for searches on encrypted data. In *Proceedings of the 2000 IEEE Symposium on Security and Privacy* (pp. 44-55). IEEE.

[118] Soror, A. A., Minhas, U. F., Aboulnaga, A., Salem, K., Kokosielis, P., & Kamath, S. (2008). Automatic virtual machine configuration for database workloads. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. ACM.

[119] Transaction Processing Performance Council. (2010). *TPC Benchmark C standard specification, revision 5.11*.

[120] Transaction Processing Performance Council. (2018). *TPC Benchmark H standard specification, revision 2.18*.

[121] Tu, S., Kaashoek, M. F., Madden, S., & Zeldovich, N. (2013). Processing analytical queries over encrypted data. *Proceedings of the VLDB Endowment, 6*(5), 289-300.

[122] U.S. Department of Health and Human Services. (2013). *HIPAA Security Rule*, 45 CFR Part 164, Subpart C.

[123] Uchechi, N., Ominyi, M., Anichukwueze, C. C., & Jones, B. C. (2019). Corporate policy advocacy and regulatory reform: Streamlining statutory bidding workflows in emerging energy markets. Iconic Research and

Engineering Journals, 3(5), 672-709. https://doi.org/10.64388/IREV3I5-1722791

[124] Unamma, U. M.-L., Okwesa, I. J., Iwuanyanwu, U., & Ajiga, D. (2018). An offline-first design framework for digital learning in low-connectivity environments. Iconic Research and Engineering Journals, 2(3), 192-211. https://doi.org/10.64388/IREV2I3-1722627

[125] Webster, J., & Watson, R. T. (2002). Analyzing the past to prepare for the future: Writing a literature review. *MIS Quarterly, 26*(2), xiii-xxiii.

[126] Wikiri, M. (2019). Global technology talent development in emerging markets: A conceptual framework for preparing software professionals for international employment. Iconic Research and Engineering Journals, 3(5), 651-671. https://doi.org/10.64388/IREV3I5-1723088

[127] Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE)*, Article 38. ACM.

[128] Zhu, B., Li, K., & Patterson, H. (2008). Avoiding the disk bottleneck in the Data Domain deduplication file system. In *Proceedings of the 6th USENIX Conference on File and Storage Technologies (FAST)*. USENIX Association.