Differential Privacy in Dynamic Encryption: Balancing Security and Utility in Cloud-Based Systems

Yunkuan Liu*

Informatics Department, University of California, Irvine, Irvine, CA 92697-1075, USA

Abstract. With the rapid development of technologies, treating data with confidentiality protection for cloud systems treating data with the guarantee of confidentiality and integrity, so that analytics are meaningful and responsive, is inevitable. The paper argues that Differential Privacy (DP) and Dynamic Encryption (DE) are two promising complementary approaches to secure the cloud system. Specifically, DP codes inference risk from released results while DE mitigates unauthorized access with fine-grained encryption, key rotation, and limited ciphertexts computation. To be more precise, the paper proposes a deployable using both DP and DE reference architecture for the cloud data platform, with a purpose that realizing the optimization of utility–latency–security. Then experimental evaluation based on aggregate Structured Query Language (SQL) queries and Differentially Private Stochastic Gradient Descent (DP-SGD) training will be briefly outlined. Recent work shows that hour-scale (or continuous) key rotation with moderate privacy budgets is able to provide a pragmatic Pareto balance for common query-heavy applications. Specifically, more homomorphic operations can be applied on low depth aggregates to improve security. By merging these two totally different but beneficial fields, it is believed that an insightful view will be explored on the development of secure-by-design and practically meaningful cloud data platforms.

1 Introduction

According to Gartner, data migration from businesses to public cloud environment has currently turn into a hot trend and there is a conflict between strong security requirement and data analytics value [1]. Following these, two emerging technologies rapidly rise in this field of study. Firstly, DP is mature from a theoretical concept to a practical framework for measuring the privacy loss of data analysis and machine learning [2]. Secondly, DE includes cryptographic methods where the protective measures change over time. That is, current key rotation, Attribute-Based Encryption (ABE) for granular based based access and Homomorphic Encryption (HE) are accessible secure computation [3]. Hence, this paper believes that a combination of DP and DE is important in modern cloud data platforms to deal with sensitive data.

* Corresponding author: yunkual@uci.edu

2 Related Work

2.1 Differential privacy

Differential Privacy provides a mathematical definition of privacy loss, and according to this definition, the presence or absence of any single individual has a negligible impact on the output of any data processing algorithm. Not only that, DP is applied in federated learning frameworks as an extra layer of protection in distributed settings where data never leaves the device [4]. In practice, DP may be challenging to implement in industry due to tooling issues and difficult privacy accounting, which necessitates an understanding of the utility-privacy trade-off, particularly in the setting where lives may be impacted, such as clinical prediction models [5].

2.2 Dynamic encryption

Dynamic Encryption refers to cryptographic systems that are designed for adaptability and ongoing security issues. To elaborate, the ABE is defined as a transformative technology that allows data to be encrypted under a policy, rather than tailoring to a specific user's public key. Moreover, Searchable Encryption extends on this concept by enabling users to search over encrypted data indices without the need to decrypt them first. This reveals a vital capability for log analysis and forensic investigations, especially in regulated industries [6].

Additionally, HE allows specific types of computations to be performed directly on ciphertexts. For analytics, HE is found to be useful for simple aggregates such as sums and means. Nonetheless, HE could incur significant computational overhead. In terms of the latency and cost under realistic cloud settings [7]. The choice of scheme and parameters would have a profound impact on subsequent performance and capabilities [8]. Moreover, traditional periodic key rotation may cause significant Input/Output bottlenecks during bulk re-encryption. Current studies have focused on developing more efficient protocols. Thus, effective key management of generation, distribution, and rotation aspects is critical to system security [9]. In constrained environments such as the Industrial Internet of Things (IIoT), key rotation strategies must be carefully designed and managed so that device capabilities are aligned with secure boot processes [10].

More dynamic encryption techniques have been developed to further enhance security. For instance, algorithm switching or cryptographic agility is a critical design principle for modern systems in response to newly discovered vulnerabilities or evolving security standards [11].

2.3 A hybrid approach

There are limitations of each approach when implemented in isolation. To be specific, DE effectively reduces risks of unauthorized access owing to key leaks, snapshot theft, or insider threats with excessive privileges. However, it fails to address the privacy risk that arises when an authorized user performs an analysis whose results cause information leakage about individuals in the dataset. DP directly addresses this "output leakage" while not protecting the underlying private data. In multi-tenant cloud analytics environments treating sensitive data, a hybrid approach that combines DP and DE can be a more comprehensive way to ensure modern security and privacy baselines without paralyzing further analysis [1].

3 Methodology

3.1 Threat, system, and governance model

With respect to external attackers, they are defined as entities intending to exfiltrate plaintext data by compromising cloud credentials, exploiting software vulnerabilities, or stealing key material. Moreover, inference adversaries refer to parties with legitimate access to query interfaces or model outputs who attempt to infer sensitive information and private data, such as through membership inference or attribute inference attacks. Furthermore, malicious insiders are authorized users who abuse their broad data access entitlements for unauthorized purposes.

The practical security concerns about cloud-based systems are related to uncertainties about internal and external threats [1]. To elaborate, the proposed architecture targets a typical cloud data platform comprising object storage, a columnar data warehouse, Extract, Transform, Load (ETL)/ Extract, Load, Transform (ELT) pipelines, and machine learning training workloads. In this case, The data sensitivity is analogous to healthcare or Internet of Things (IoT) telemetry data, whereas access patterns include both scheduled batch jobs and ad-hoc interactive queries. Meanwhile, a dual-pronged governance approach is essential for operations to be effectively carried out. In particular, a centralized privacy budget service enforces per-dataset privacy caps (ϵ-budgets) as an immutable ledger of all data releases (recording the mechanism, ϵ consumed, and requester), offering pre-approved DP mechanisms for common tasks [2]. And cryptographic governance requires a robust key lifecycle management system that are integrated with a hardware security module or cloud key management system to manage root keys. Key rotation policies are defined per data partition, and comprehensive audit logs track all cryptographic operations, which are alignsed with national institute of standards and technology key management guidelines and best practices for cryptographic enforcement [9, 10].

3.2 The differential privacy layer

This differential privacy layer is responsible for adding calibrated noise into outputs for a formal privacy guarantee. For counting and aggregation queries, f(D), with a global sensitivity Δf, which signifies the maximum change in the output given any single record change, the Laplace Mechanism is expressed below:

$$\tilde{f}(D) = f(D) + Lap\left(\frac{\Delta f}{\epsilon}\right) \tag{1}$$

$f(D)$ represents the true, non-private result of a query (e.g., a COUNT or SUM) on the dataset D, where Δf (Global Sensitivity) denotes the maximum amount a single individual's data can change the query's result and determines how much noise is needed, ε (Epsilon / Privacy Budget) controls the privacy-accuracy tradeoff such that a smaller ε provides stronger privacy by adding more noise while reducing the accuracy of the output, and Lap() indicates that the Laplace distribution is the source of the random noise added to the true result.

For machine learning training, the DP-SGD is applied, including clipping the gradient of each training example to a fixed norm and adding Gaussian noise to the average gradient during each update step. The total privacy loss in this case is denoted as ϵ total， which is shown over T training steps. This is carefully computed via advanced composition theorems such as the Renyi Differential Privacy [2]. Here, σ stands for the noise multiplier, q stands for the sampling rate, and δ stands for a small probability of failure.

$$\epsilon_{total} \approx \mathcal{A}(\sigma, q, T, \delta) \tag{2}$$

The DP-SGD Privacy Accounting represents the privacy accounting for the DP-SGD algorithm applied in machine learning training, where σ (Noise Multiplier) controls the specific amount of Gaussian noise added to the gradients during training with higher values indicating more noise and better privacy, q (Sampling Rate) represents the fraction of the dataset used in each training step (mini-batch size / total dataset size), T (Training Steps) denotes the total number of steps in the training process, and δ (Delta) is a small probability that the privacy guarantee might fail, typically set to a value less than the inverse of the dataset size, while A() serves as the privacy accounting function that composes the privacy loss across all training steps to calculate the total ε. In this case, the privacy budget is strategically allocated and split into $\epsilon=\epsilon_{critical}+\epsilon_{routine}$, where a cap is placed on $\epsilon_{critical}$ for high-impact releases to ensure long-term protection of privacy data.

3.3 The dynamic encryption layer

The dynamic encryption layer can be categorized into three primary functional planes. First, for storage encryption and dynamic key generation, data at rest will be encrypted using standard symmetric algorithms. Access-pattern-based key generation is a strategy aiming to provide more responsive security, where keys can also be regenerated based on specific access patterns. For example, a key change may be triggered after a certain number of data access events, upon access from a new geographic location, or following the invocation of high-risk administrative Application Programming Interfaces. This context-aware approach, inspired by risk-based authentication systems, ensures that the cryptographic layer responds dynamically to potential threat indicators [12].

Second, ABE is applied to enforce complex access policies at the cryptographic level. To elaborate, if a user's attributes are revoked, the affected data segments will be automatically re-encrypted. Therefore, searchable encryption allows for secure indexing and searching over encrypted log data or other textual fields without the need for decryption [6]. Third, for a subset of low-depth analytical operations, HE schemes can be employed. The parameters for this particular scheme should be carefully chosen based on the computational depth required and corresponding latency constraints [3, 7].

3.4 Multi-objective optimization formulation

To determine configurations that optimally balance competing objectives. Let $U(\epsilon,\tau,\theta)$ denote utility, $L(\epsilon,\tau,\theta)$ denote system latency, and $S(\epsilon,\tau,\theta)$ denote a composite security score, which combines a privacy risk component and an access risk component. For a given workload W and cryptographic parameter set θ, the following equation will be solved:

$$\max_{\epsilon,\tau,\theta} \Phi = w_U U - w_L L \quad subject\ to \quad S \geq S^*, \epsilon \in [\epsilon_{min}, \epsilon_{max}], \tau \in \mathcal{T} \tag{3}$$

This multi-objective optimization formulation is the core of the proposed hybrid approach, aiming to maximize a weighted combination of utility (U) and the negative of latency (L), thereby seeking to achieve high utility and low delay. In particular, it will be performed under the critical constraint that the overall security score (S) never falls below a minimum acceptable threshold (S*), so that security and privacy will never compromise the overall performance.

3.5 Evaluation metrics

For Analytics Workload: Relative Root Mean Square Error (RRMSE), the formula below evaluates the accuracy of aggregate queries under Differential Privacy, measuring the relative error introduced by the noise mechanisms:

$$RRMSE = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \hat{Y}_i\right)^2}}{\bar{Y}} \tag{4}$$

N: The total number of test queries executed.
Y_i: The true result of the i-th query (without DP noise).
Ŷ_i: The noisy, privacy-protected result of the i-th query (released by the DP layer).
Ȳ: The mean of the true results across all N queries.

In this case, the RRMSE quantifies the relative magnitude of the error introduced by the DP noise. A lower RRMSE indicates higher utility and better accuracy, as the noisy results are closer to the true values. The denominator normalizes the error, so that it is comparable across queries of different scales.

To assess privacy risk for security measures, Membership Inference Attack Success Rate (ASR) evaluates the practical privacy guarantee by testing the resistance of the system against a specific privacy attack:

$$ASR = \frac{Number\ of\ Successful\ Inferences}{Total\ Number\ of\ Inference\ Attempts} \tag{5}$$

When an adversary correctly determines if a specific record was part of the model's training dataset. In other words, a lower ASR indicates a stronger effective privacy protection. A rate close to 50% (random guessing for a binary inference) is the target, showing that the DP and encryption layers successfully prevent leakage of membership information.

To assess Time-at-Risk (TAR), this metric precisely evaluates the effectiveness of the dynamic encryption layer in limiting the exposure window of a compromised cryptographic key:

$$TAR = Key\ Compromise\ Time - Key\ Rotation\ Time \tag{6}$$

Key Compromise Time: The hypothetical moment a key is stolen by an adversary.

Key Rotation Time: The next scheduled or policy-triggered time when the key is retired and data is re-encrypted with a new key.

A shorter TAR signifies a more responsive and secure cryptographic system. It measures the maximum potential window for data exfiltration after a key is compromised. The dynamic key generation policies aim to minimize this window, with continuous rotation theoretically reducing TAR to near zero.

System Performance can be measured by P95 Latency for Queries (L_p95), showing the end-to-end responsiveness of the system from a user's perspective:

$$L_{p95} = P_{95}(\{l_1, l_2, \ldots, l_N\}) \tag{7}$$

l_i: The latency (response time) of the i-th query in the workload.
P_95(): The 95th percentile function.
To compute the Encryption/Re-encryption Throughput (Θ):

$$\Theta = \frac{Total\ Data\ Volume\ Processed\ (MB)}{Total\ Processing\ Time\ (s)} \tag{8}$$

In this case, throughput, measured in megabytes per second (MB/s), quantifies the system's capacity to perform encryption and lazy re-encryption operations. A higher throughput indicates that the security layer can operate with minimal impact on data ingestion and management pipelines.

Moreover, the Storage Overhead (Ω) quantifies the additional storage cost incurred by the cryptographic and privacy-preserving techniques:

$$\Omega = \frac{S_{protected} - S_{original}}{S_{original}} \times 100\% \tag{9}$$

S_original: The size of the original, plaintext dataset.
S_protected: The total size of the protected dataset, including encrypted data, ABE metadata, and DP noise parameters.

Expressed as a percentage, this metric captures the bloat introduced by encryption (e.g., ciphertext is often larger than plaintext) and any metadata required for ABE or managing DP budgets. In particular, a lower overhead is desirable to control cloud storage costs.

4 Results and Analysis

4.1 Discussion, workload, factors

Findings suggest that data utility decreases as the privacy guarantee strengthens (lower ϵ). For aggregate queries, the RRMSE remains below 5% for $\epsilon \geq 2$, explaining why it is suitable for many analytical purposes. For the Machine Learning (ML) task, the model's Area Under the Curve (AUC) remains high (>0.88) for $\epsilon \geq 1$, manifesting a "knee" in the trade-off curve within the $\epsilon \in [1,2]$ range. This stays in line with findings from existing studies, where this budget range demonstrates a balance between statistical usefulness and privacy protection [5].

The evaluation is conducted through the UCI Adult (Census Income) Dataset, a tabular dataset containing sensitive attributes such as level of income. Two representative workloads are defined: analytics workload refers to a series of SQL-like aggregation queries (COUNT, SUM, AVG) with filters and group-by clauses, simulating typical business intelligence tasks, whereas machine learning workload trains a binary classifier, such as predicting the level of income level, using DP-SGD, with AUC used as the primary utility metric.

Experimental Factors:
Privacy Budget (ϵ): {0.5,1.0,2.0,4.0}
Rotation Cadence (τ): {24h,12h,1h,10m,continuous}

Crypto Stack: (i) Advanced Encryption Standard- Galois/Counter Mode (AES-GCM) only (baseline), (ii) AES-GCM + ABE, (iii) AES-GCM + ABE + HE for aggregates.

Baselines: DP-only, DE-only, and the proposed Hybrid (DP+DE) approach.

Table 1. The utility-privacy trade-off for different workloads with hypothetical data.

Privacy Budget (ε)	RRMSE for Analytics (%)	AUC for ML Model
4.0	2.3	0.90
2.0	4.8	0.88
1.0	7.9	0.86
0.5	13.7	0.82

Table 1 quantifies the core trade-off between the privacy budget (ε) and data utility. It clearly shows that as privacy strengthens (ε decreases), utility worsens (RRMSE increases, AUC decreases). The row for ε=2.0 is highlighted to visually demonstrate the "knee" in the trade-off curve, where a small decrease in ε still maintains high utility.

For queries with integer outputs or low sensitivity, the Laplace mechanism is suitable. However, for high-dimensional queries or machine learning gradients, the Gaussian mechanism is usually preferred on account of its distinct composition properties. Utilizing the Gaussian mechanism for the DP-SGD workload would result in a marginally better utility, as shown by approximately 2 to 3% higher AUC for the same ϵ value compared to the Laplace mechanism, at the cost of introducing a negligible δ term. These findings are consistent with existing studies [2]. In this case, the computational overhead of generating Gaussian noise versus Laplace noise has been negligible in this paper.

The privacy-utility tradeoff is demonstrated across different epsilon values, where $\varepsilon = 4.0$ provides high utility but a weaker privacy guarantee, $\varepsilon = 2.0$ represents the "knee" of the curve and offers an optimal balance for many use cases, $\varepsilon = 1.0$ achieves a balanced state suitable for most analytical tasks requiring strong privacy, and $\varepsilon = 0.5$ delivers strong privacy but with significantly reduced utility.

Table 2. Performance and security impact of key rotation cadence.

Rotation Cadence (τ)	P95 Query Latency (ms)	Relative Latency Increase	Avg. TAR
24 hours	120 ms	Baseline (0%)	24 hours
12 hours	132 ms	+10%	12 hours
1 hour	144 ms	+20%	1 hour
Continuous	138 ms	+15%	< 1 minute

Table 2 displays the cost-benefit analysis of the Dynamic Encryption layer. It shows that faster key rotation (moving down the table) improves security by drastically reducing the Time-at-Risk window. Nevertheless, this comes at a performance cost, as seen in the increased P95 latency. The row for "Continuous" rotation indicates success of the throttled update strategy, offering the best security while mitigating the extreme latency penalty seen with the hourly "bulk update."

In addition, another important finding was that the performance impact of the DP and DE layers was largely additive rather than multiplicative. In other words, the latency of a query protected with both DP and time-based key rotation was approximately the sum of the latency introduced by each layer independently, which means that the two technologies can be integrated without introducing unexpected synergistic performance bottlenecks, which serves as a positive indicator for the feasibility of the hybrid architecture.

Overall, the hybrid DP+DE configuration consistently outperforms the single-method baselines on the composite security score. To be more specific, DP-only reduces inference

success but leaves data exposed to access breaches. DE-only reduces the exposure window but fails to effectively prevent leakage from outputs. The hybrid approach, particularly at ϵ=1 with 1-hour or continuous rotation, simultaneously reduces membership inference success and bounds the time-at-risk to a short window.

Table 3 effectively synthesizes why the hybrid approach is superior. It shows that individual methods only solve one part of the problem: DP-only mitigates inference attacks but not key exposure, while DE-only reduces the exposure window but not inference risks. The hybrid approach successfully combines both strengths, leading to a low inference success rate and a bounded Time-at-Risk, which results in the highest composite security score.

Table 3. Comparative security efficacy of architectural configurations.

Architecture Configuration	Membership Inference Success Rate	Time-at-Risk	Composite Security Score
Baseline (No DP/DE)	85% (Very High)	∞ (Persistent)	20 / 100
DP-Only (ε=1)	55% (Near Random)	∞ (Persistent)	60 / 100
DE-Only (τ=1h)	70% (High)	1 hour	65 / 100
Hybrid (DP+DE) (ε=1, τ=1h)	56% (Near Random)	1 hour	90 / 100

4.2 Practical deployment

Operationalizing the proposed architecture needs to address practical challenges. In particular, privacy budget accounting is critical; that is, without a centralized ledger and automated enforcement, budgets can be easily exceeded [2]. For key rotation, the choice between periodic and continuous strategies depends on workload patterns and risk tolerance. Continuous rotation is preferable for highly sensitive, actively queried data; however, it must be implemented with care to manage resource consumption [10]. Furthermore, homomorphic encryption should be deployed selectively for specific, high-value aggregates where its large computational cost is justified, leveraging optimizations identified in performance studies [3,7].

4.3 Design guidelines

For publicly released statistics, DP should be prioritized. By contrast, for internal dashboards with sensitive data, DP can be used for shared extracts and DE for internal access control. To set the budget, $\epsilon \in [1,2]$ is marked as a starting point for interactive analytics, and validation curves are applied to tune ϵ for ML models based on acceptable utility loss [5]. In the meantime, an hour-scale or continuous key rotation should be implemented as a security baseline to effectively manage sensitive datasets [9]. In addition, HE will be limited to low-depth computations and profile performance thoroughly via guidance from recent evaluations [7, 8]. And privacy ledgers will be conducted for budget accounting, attributing catalogs for ABE, and automating key lifecycle management to ensure long-term security and compliance [9].

5 Conclusion

This paper demonstrates that Differential Privacy and Dynamic Encryption serve as complementary rather than competing strategies for securing cloud-based systems. The proposed hybrid architecture effectively integrates both approaches, achieving a balanced trade-off between utility, performance, and security across common cloud workloads. Experimental results show that moderate privacy budgets ($\varepsilon \in [1,2]$) combined with hour-scale or continuous key rotation provide practical security guarantees while maintaining acceptable system performance and data utility. The hybrid approach consistently outperforms single-method baselines, simultaneously reducing membership inference success rates to near-random levels while bounding the time-at-risk window to short intervals. For organizations deploying cloud analytics with sensitive data, this work provides actionable guidance: prioritize DP for public releases, implement hour-scale key rotation as a security baseline, and selectively apply homomorphic encryption to low-depth operations where justified by security requirements.

Future research should extend this framework to unstructured data sources such as multimedia streams and IoT telemetry, investigate adaptive resource scheduling to reduce computational overhead from continuous key rotation, and explore hardware acceleration techniques to enable real-time analytics under strong privacy guarantees. The path forward for secure cloud analytics requires not only algorithmic innovation but also practical, scalable system integration that balances competing objectives in production environments.

References

1. N.H. Mohamad, N.B. Saidin, M.I.H.B. Zaidi, Data Security and Privacy Issues in Cloud Computing: Challenges and Solutions Review. TechRxiv (2023)
2. M. Kilpala, T. Kärkkäinen, T. Hämäläinen, Differential Privacy: An Umbrella Review, in Artificial Intelligence and Cybersecurity, Springer, Cham (2023), pp. 167-183
3. C. Juvekar, V. Vaikuntanathan, A. Chandrakasan, Gazelle: A Low Latency Framework for Secure Neural Network Inference. arXiv **1801.05507** (2018)
4. M. El Moudni, E. Ziyati, A Hybrid Approach to Enhancing Personal Sensitive Information Protection in the Context of Cloud Storage. International Journal of Computing and Digital Systems **17**, 1-14 (2025)
5. A. Dyda, M. Purcell, S. Curtis, et al., Differential privacy for public health data: An innovative tool to optimize information sharing while protecting data confidentiality. Patterns **2**, 100366 (2021)
6. S. Wang, S. Jia, Y. Zhang, Verifiable and Multi-Keyword Searchable Attribute-Based Encryption Scheme for Cloud Storage. IEEE Access **7**, 50136-50147 (2019)
7. J.H. Cheon, A. Costache, R.C. Moreno, et al., Introduction to Homomorphic Encryption and Schemes, in Protecting Privacy through Homomorphic Encryption, Springer, Cham (2021), pp. 3-28
8. A. Aloufi, P. Hu, H.W.H. Wong, S.S.M. Chow, Blindfolded Evaluation of Random Forests with Multi-Key Homomorphic Encryption. IEEE Transactions on Dependable and Secure Computing **18**, 1821-1835 (2021)
9. E. Barker, W.C. Barker, Recommendation for Key Management: Part 2 – Best Practices for Key Management Organizations. NIST Special Publication 800-57 Part 2 Revision 1 (2019)
10. S. Deshpande, R. Jogdand, A Survey on Internet of Things (IoT), Industrial IoT (IIoT) and Industry 4.0. International Journal of Computer Applications **175**, 20-27 (2020)
11. C. Ma, L. Colon, J. Dera, B. Rashidi, V. Garg, CARAF: Crypto Agility Risk Assessment Framework. Journal of Cybersecurity **7**, tyab013 (2021)

12. A. Gunuganti, Risk Based Authentication. Journal of Artificial Intelligence & Cloud Computing **1**, 1-6 (2022)