Adaptive Quantum-Resistant Hybrid Encryption Framework for Secure IoMT Edge Data Sharing

Zahra Seyedi, Department of Mathematics and Computer Science, Amirkabir University of Technology,
Stefan Rass, Johannes Kepler University Linz, LIT Secure and Correct Systems Lab,
Shahzad Ahmad, Johannes Kepler University Linz, LIT Secure and Correct Systems Lab,
Farhad Rahmati, Department of Mathematics and Computer Science, Amirkabir University of Technology,

Abstract—The Internet of Medical Things (IoMT) is revolutionizing healthcare through real-time monitoring and personalized care, yet existing security solutions for resource-constrained IoMT devices often compromise between robust protection and computational efficiency. Additionally, various IoMT systems utilize different sets of encryption schemes, ranging from classical to post-quantum algorithms, tailored to their computational resources, operational constraints, and specific privacy challenges. This diversity highlights an urgent need for an adaptive and efficient, quantum-resistant security protocol applicable to these systems. We propose the Privacy-Preserving Hybrid Data Sharing (P^2HDS) framework, which integrates multiple classical and post-quantum encryption schemes to achieve strong security with enhanced efficiency. P^2HDS formalizes a generalized hybrid encryption protocol that enables adaptive configuration based on security requirements and computational constraints. We employ game-theoretic analysis to model interactions between data owners and adversaries, determining the optimal combination of encryption schemes for data outsourcing to mitigate adversarial threats. Rigorous security analysis proves that P^2HDS achieves indistinguishability under chosen-plaintext attack (IND-CPA) security in the random oracle model, provided at least one constituent scheme is IND-CPA-secure. Experimental results demonstrate that P^2HDS delivers performance approximating the average of its constituent schemes while maintaining security comparable to the most secure component, making it a scalable, quantum-resistant, and adaptive solution for privacy-preserving IoMT edge applications.

Index Terms—hybrid encryption, all-or-nothing transformation, public key encryption, Internet of Medical Things.

I. Introduction

The advancement of technology is notably transforming various industries, particularly healthcare. The Internet of Medical Things (IoMT) is enhancing healthcare by making it more proactive, cost-effective, and personalized. It focuses on key areas like disease prevention, health monitoring, and diagnosis [1]. According to MarketsandMarkets, the global IoMT market is expected to grow from $127.7 billion in 2023 to $289.2 billion by 2028, with a CAGR of 17.8% [1]. This growth is driven by the rising use of wearable devices, demand for remote patient monitoring, and the prevalence of chronic diseases. Telemedicine is predicted [2] to dominate the IoMT market, particularly following its critical role during the COVID-19 pandemic in facilitating remote consultations and patient monitoring while minimizing physical contact [1].

Achieving effective health monitoring requires extensive computing and communication resources, which can be inefficient for mobile devices due to their limited capabilities and battery life. A centralized cloud-based IoMT framework is often employed to address these limitations by collecting healthcare data from various devices and offloading it to cloud systems like medical centers for processing [2]. However, during large-scale health crises like COVID-19, the surge in patients can overwhelm this centralized approach. This is due to increased latency from the cloud's distance from users and the risk of a single point of failure, which can disrupt access to vital healthcare services [3]. Moreover, cloud storage may not be suitable for applications requiring latency under a few tens of milliseconds, as heavy data requests can lead to network congestion and negatively impact user experience [4, 5]. Edge Computing (EC) offers a promising way to lighten the load on Edge Servers (ESs) and medical centers. When combined with the IoMT, it allows processing of data from body sensors at ESs. Patient information and location data from ESs are sent to medical centers, reducing computational demands and transmission delays for real-time analysis. Additionally, because data is processed at the network's edge, experts can remotely access and manage it using various devices [3, 6].

The security of IoMT systems is a significant challenge, demanding robust protection for highly valuable patient data, which costs far more than credit card information and directly impacts lives [7, 8]. In this context, encryption methods are a fundamental solution for protecting sensitive data. EC then acts as a complement. By decentralizing computation, EC enables DOs to distribute encrypted sensitive data across various edge servers to enhance privacy, optimize resources, and boost resilience. However, this distributed architecture introduces new risks like collusion among adversaries and unauthorized access. Current security approaches often fall into two extremes: either they rely on bandwidth-intensive, highly secure protocols, or they use lightweight, less secure methods. This dichotomy highlights a significant gap in finding solutions that effectively bridge the critical divide between strong security and efficient performance in IoMT edge

[1] https://www.marketsandmarkets.com/Market-Reports/iot-healthcare-market-160082804.html

[2] https://www.alliedmarketresearch.com/internet-of-medical-things-market-A07917

environments.

Inspired by the approaches introduced by Rass et al. [9, 10], we address this challenge with a Privacy-Preserving Hybrid Data Sharing (P^2HDS) scheme, a general framework that combines multiple encryption schemes, $\Pi_1, \ldots, \Pi_n$, into a single scheme. Provided at least one of these encryption schemes is strongly secure, P^2HDS ensures security nearly as strong as the secure one. Crucially, it should be significantly more efficient and far less bandwidth-intensive than those heavy, perfectly secure methods. Our main contributions are as follows.

- Formalization of the P^2HDS Framework: We propose and formalize a generalized framework for secure data sharing in edge computing environments. The scheme takes as input a set of encryption schemes with varying security levels, ranging from lightweight to highly secure, and a set of edge servers, producing a unified hybrid encryption protocol that leverages the strengths of its constituent schemes to ensure robust privacy and efficient resource utilization.
- Game-Theoretic Analysis of the Outsourcing Method: We model the interaction between a data owner and a potential adversary as a strategic game to optimize the selection outsourcing configuration in P^2HDS scheme. This analysis identifies the data owner's optimal strategy and strengthens resilience against adversarial threats in the proposed edge-based data sharing environment.
- Security Analysis: We provide an extensive security proof demonstrating that P^2HDS achieves Indistinguishability under Chosen-Plaintext Attack (IND-CPA) security in the random oracle model, provided at least one of the underlying encryption schemes is IND-CPA-secure. The proof establishes that the security of P^2HDS is comparable to that of the most secure constituent scheme.
- Efficiency and Security Evaluation: Experimental results demonstrate that the performance of P^2HDS approximates the average of its constituent encryption schemes, achieving a balance of computational and bandwidth efficiency. While maintaining security comparable to the most secure scheme, P^2HDS reduces resource demands compared to highly secure but computationally intensive protocols.

A. Paper Organization

The rest of this paper is structured as follows. Section II reviews related work. Section III formulates the problem, detailing the system model, P^2HDS definition, threat model, and security requirements. Section IV describes the P^2HDS approach by detail. Section V provides a game-theoretic analysis of data outsourcing structure and configuration selection. Section VI presents the security analysis and Section VII provides the performance analysis. Finally, Section VIII concludes the paper.

II. Related Work

This section provides a literature review for this study. We first explore the existing research on hybrid encryption schemes, followed by a brief discussion of the all-or-nothing transformation.

A. Hybrid Encryption

The concept of combining multiple cryptographic primitives to enhance security traces back to Shannon's foundational work on the "product" of secrecy systems [11]. This idea of a sequential construction, often termed a cascade cipher, was a primary focus of early research into block ciphers [12, 13], culminating in the development and security analysis of Triple DES (Data Encryption Standard) [14, 15]. In modern cryptography, this strategy is often referred to as multiple encryption, combiners, or, more broadly, hybrid encryption. The principle behind hybrid encryption is to integrate various independent schemes, frequently grounded in distinct computational assumptions, to form a new, more robust cryptographic construction [16]. While some constructions reuse the same scheme with different keys, modern approaches often combine schemes based on different computational assumptions, particularly in the transition to post-quantum cryptography (PQC).

This strategy ensures resilience against both classical and quantum adversaries, making hybrid schemes a critical area of research in PQC, as evidenced by comprehensive studies like [17, 18, 19, 20], which underscore the rising interest in hybrid PQC protocols, especially for key encapsulation mechanisms (KEMs) and digital signatures. Giacon et al. [18] studied classical-secure hybrid KEMs, and Bindel et al. [19] adapted security models for quantum adversaries. Huguenin-Dumittan and Vaudenay [20] generalized the Fujisaki-Okamoto transform to develop efficient post-quantum KEM combiners. The systematic review by Giron et al. [17] highlights the emphasis on post-quantum hybrid key exchange methods.

Despite this progress, hybrid asymmetric encryption has been relatively overlooked compared to hybrid KEMs and digital signatures, particularly in the PQC landscape. Within the domain of public-key cryptosystems, one of the earliest contributions is the work of Asmuth and Blakley, who proposed using the XOR operator to mask a plaintext message with multiple individual ciphers [21]. Herzberg [22] later analyzed the security of various "folklore" combiners, such as the XOR-input combiner and the sequential public-key construction. However, a significant drawback of these early, natural constructions is their vulnerability to adaptively chosen-ciphertext attacks (IND-CCA). This weakness arises from their inability to validate malicious ciphertexts during decryption. To address this, Zhang et al. [23] and Dodis and Katz [24] independently proposed IND-CCA-secure asymmetric multiple encryption schemes. Zhang et al. leveraged hash functions within the random oracle model (ROM) to construct security from weaker components, while Dodis and Katz employed additional tools like secret sharing and digital signatures to achieve IND-CCA security in the standard model.

Subsequent advancements by Fujioka et al. [25] offered a more efficient IND-CCA-secure sequential cipher,

closely resembling natural combiners, and Hohenberger et al. [26] introduced a detectable-CCA (IND-dCCA) notion to bridge weaker components to IND-CCA security, albeit with the caveat that each component must remain secure. More recently, Goncalves and Mashatan [27] pioneered the first public-key multiple encryption scheme secure in the quantum random oracle model (QROM), marking it as the only hybrid scheme analyzed across both ROM and QROM frameworks to date.

Leading tech companies have also advanced hybrid post-quantum schemes. In 2016, Google began integrating a hybrid key exchange in Chrome, combining elliptic curve Diffie-Hellman (ECDH) with the NewHope PQC scheme [28, 29]. In 2018, it paired the HRSS PQC scheme [30] with ECDH to secure its internal Application Layer Transport Security [31]. Other companies like Cloudflare [32], Microsoft Research [33], and Amazon [34] are also developing hybrid schemes, highlighting the practical significance of this research. However, a notable limitation of these existing schemes is their reliance on predefined combinations of specific encryption algorithms to construct hybrid systems, tailored to particular security requirements or post-quantum readiness. None of these approaches offer a generalized framework that empowers a system or the data owner to flexibly select and integrate their own accessible encryption schemes, whether classical, post-quantum, or a mix thereof, to construct a hybrid scheme customized to their desired security level. This gap highlights the need for a flexible, user-driven solution that allows the dynamic combination of multiple encryption schemes to achieve a security profile optimized for specific use cases, addressing both efficiency and resilience against evolving threats, which is our main goal in this paper.

B. All-or-Nothing Transforms

The All-or-Nothing Transformation (AONT) was first introduced by Rivest [35] as a cryptographic primitive that ensures an encrypted message cannot be partially decrypted, either the entire message is recoverable, or nothing is revealed, ensuring that an adversary must recover the entire ciphertext to gain any information about the plaintext. The proposed scheme transforms plaintext into multiple blocks through a preprocessing step involving symmetric encryption and hash-based dependencies, followed by a second encryption layer, though it remains vulnerable if the initial random key is exposed alongside the final encryption key. Boyko [36] further analyzed the security of AONTs in different adversarial models, strengthening their theoretical foundations. Desai [37] also enhanced this concept by replacing hash computations with a faster XOR operation of the random key with prior blocks.

Recent innovations include Bastion's approach [38], which applies a linear transform to encrypted data to ensure that the initial ciphertext blocks cannot be recovered as long as the adversary has access to all but two output blocks. In [39] also, an approach called Mix&Slice is introduced for cloud storage, creating inter-block dependencies via multiple encryption rounds, enabling access revocation by re-encrypting small data portions with a new key, a feature akin to AONT's requirement for the full ciphertext. The Secure Fragmentation and Dispersal (SFD) scheme [40] builds on Secret Sharing Made Short (SSMS) and AONT, performing encryption and fragmentation separately in a coarse-grained manner, prioritizing data dispersal over block dependencies. The Circular All-Or-Nothing (CAON) scheme [41] refines Bastion's technique, halving the linear transform's complexity by chaining blocks with XOR operations, fragmenting across sites for full recovery. Additionally, a recent work in [42] explores the security of combinatorial AONTs, building on Rivest's and Stinson's foundations. This study quantifies the information leakage, establishing bounds using information-theoretic methods, showing perfect security with equiprobable inputs and weak security with non-zero probability inputs, bridging the gap between these security levels.

Overall, AONTs have emerged as powerful tools in cryptography, offering keyless protection against partial data disclosure, seamless integration with lightweight and hybrid encryption schemes, adaptability for bandwidth-constrained settings like IoMT and storage, and robust composability in both theoretical and practical systems. This paper leverages these strengths by incorporating an AONT preprocessing layer, enabling only a subset of components to demand perfect secrecy while relying on efficient primitives to maintain overall confidentiality.

III. Problem formulation

In this section, we first present our system model, detailing the entities involved in this EC environment, their goals in this system, along with their interaction model. Next, we outline the components and definition of the P^2HDS scheme. Finally, we describe the threat model and the associated security requirements.

A. System Model

We consider a health monitoring system for a set of patients as Data Owners (DOs) denoted as $DO = \{do_1, do_2, \ldots, do_p, \ldots, do_P\}$, a set of researchers or doctors as Data Users (DUs) indicated as $DU = \{du_1, du_2, \ldots, du_q, \ldots, du_Q\}$, a set of ESs placed closer to DUs denoted as $ES = \{es_1, es_2, \ldots, es_i, \ldots, es_J\}$, and a Central Authority (CA). Each of these components are described below.

- DOs: This entity is responsible for generating and managing medical data sets denoted as D. A DO has several body sensors. They continuously generate data files, and transmit them to the gateway. These gateways have limited computing and storage resources. To ensure confidentiality, DOs use some encryption methods, to encrypt the gathered data in gateways in compliance with a specific access structure. Then, the encrypted data set is transmitted to corresponding ESs.

- ESs: These entities are connected to access points or base stations located near DOs. These servers have moderate storage and computing capabilities. The primary duties of ESs include caching or storing the data set and furnishing nearby DUs with low latency data access.
- DUs: They refer to doctors, researchers, or other medical specialists. Each DU corresponds to a set of attributes or access policies to access the DOs' stored data on nearby ESs. So, according to these access policies, they send their data queries to the corresponding ESs.
- CA: Serving as the central processing unit, the CA is responsible for initializing the system and generating secret keys for both DOs and DUs.

In this distributed data-sharing system, after the necessary data processing, DOs relay the encrypted data to one or more nearby ESs through 5G or WiFi communication. Then, the DUs located in close proximity to these ESs can access the data with minimal delay. This entire process occurs in the following three steps:

1) Initialization: The system begins with the CA establishing a security parameter and selecting a set of encryption schemes to balance security and efficiency. It initializes these schemes, generating and sharing required parameters and keys for each DO and DU.
2) Data outsourcing: To share a dataset, a DO first divides it into multiple blocks. Then, it decides on a configuration for how these blocks will be encrypted and stored across various ESs. This configuration details the total number of blocks and specifies which encryption scheme and ES each block will use. Following this, an AONT transformation is applied to the blocks, enhancing security by making the scheme's strength dependent on the most secure underlying method. Each block is then encrypted using its assigned scheme, secretly tagged with its configuration details, and sent to the corresponding ES, ensuring retrievability by authorized DUs while maintaining confidentiality.
3) Data retrieval: An authorized DU retrieves the dataset by regenerating the configuration using its policy key, identical to the DO's setup. It queries the relevant ESs with generated tags to fetch the encrypted blocks. Each ES searches its storage based on the tag and returns the corresponding ciphertext or an error if no match is found. Once all blocks are collected, the DU decrypts them using its secret key, reconstructing the original dataset.

B. System Definition

Now, we will define the algorithms required for the described data sharing process. The P^2HDS scheme is formally defined by the following 10 Probabilistic Polynomial-time (PPT) algorithms:

- Setup$(1^\kappa) \rightarrow (msk, prm)$: When provided with the security parameter κ, it considers a set of n public key encryption schemes $\{\Pi_i\}_{i=1}^n$ and outputs the required secret and public parameters to initialize the hybrid scheme as a master secret key msk and public parameters prm. The CA is responsible for executing this algorithm.
- KG$(prm, msk, id_{du}) \rightarrow (sk^{id_{du}}, pk^{id_{du}})$: It inputs a DU's identifier as id_{du} along with public parameters prm, master secret key msk. It generates the secret key $sk^{id_{du}}$ and public key $pk^{id_{du}}$ associated id_{du}. Again, the CA is carrying out this algorithm.
- KG$_F(prm, msk, id_{do}, policy_{id_{do}}, id_{du}, A_{id_{du}}) \rightarrow (sk_F^{id_{do}})$: Given prm, msk, a DO identifier id_{do} and its associated access policies $policy_{id_{do}}$ (i.e., a set of access control rules specifying the attribute set of authorized DUs), a DU identifier id_{du} and its attribute set $A_{id_{du}}$, this algorithm returns $sk_F^{id_{do}}$. The CA performs this too.
- IdxG$(r_{max}, n, k, sk_F^{id}) \rightarrow conf = (r, \{idx_i^\Pi, idx_i^{es}\}_{i=1}^{r+1})$: It takes the maximum value r_{max} along with the number of encryption schemes and ESs, then it produces $conf = (r, \{idx_i^\Pi, idx_i^{es}\}_{i=1}^{r+1})$ as the outsourcing configuration including the number of data blocks, r, plus the indices of the encryption schemes and the indices of ESs for each block, $\{idx_i^\Pi, idx_i^{es}\}_{i=1}^{r+1}$. This can be run by a DO or a DU.
- AONT$(prm, conf, D) \rightarrow (\{t_i\}_{i=1}^{r+1})$: Given a configuration $conf = (r, \{idx_i^\Pi, idx_i^{es}\}_{i=1}^{r+1})$, it computes and returns $r+1$ transferred data blocks, $\{t_i\}_{i=1}^{r+1}$. The DO with data set D runs this.
- Enc$(conf, \{t_i\}_{i=1}^{r+1}, pk^{id_{du}}) \rightarrow (\{c_i\}_{i=1}^{r+1})$: Given the public key set $pk^{id_{du}}$, the configuration $conf$, and transferred data blocks $\{t_i\}_{i=1}^{r+1}$, it encrypt each block t_i to c_i, and returns $\{c_i\}_{i=1}^{r+1}$. The DO runs this too.
- TG$(prm, a, idx_a^\Pi, idx_a^{es}, sk_F^{id}) \rightarrow (tg_a^{id})$: This algorithm inputs an index $a \in \{1, \ldots, r+1\}$, along with its corresponding encryption scheme and ES indices from $conf = (r, \{idx_i^\Pi, idx_i^{es}\}_{i=1}^{r+1})$, means idx_a^Π and idx_a^{es}, plus the secret key sk_F^{id}, then returns a tag tg_a^{id} associated with the index a and the secret key sk_F^{id}. This can be performed by the DO or a DU.
- ESAs$(conf, \{c_i, tg_i^{id}\}_{i=1}^{r+1}) \rightarrow (\{status_i\}_{i=1}^{r+1})$: It attempts to assign each ciphered block pair (c_i, tg_i^{id}) to the corresponding ES indexed idx_i^{es} specified in $conf$, for $i \in \{1, \ldots, r+1\}$. For each assignment, it outputs a status value $status_i \in \{\text{success}, \text{failure}\}$. The DO will run this algorithm.
- Search$(tg^{id}) \rightarrow (c_i \text{ or } \perp)$: On input a tag tg^{id}, this algorithm will search for a ciphered block that is tagged tg^{id}. If it finds, it will return c_i associated with tg^{id}; otherwise, it returns $\perp$. An ES will perform this algorithm.
- Dec$(prm, \{c_i\}_{i=1}^{r+1}, sk^{id}) \rightarrow D$: It decrypts ciphered blocks $\{c_i\}_{i=1}^{r+1}$ and retrieves $\{t_i\}_{i=1}^{r+1}$, then recovers d_i for $i = \{1, \ldots, r\}$. It outputs $D = d_1 || \cdots || d_r$.

Definition 1. A P^2HDS scheme is considered correct if, for given encryption schemes $\{\Pi_i\}_{i=1}^n$, security parameter κ, DOs and DUs set, and a Data set D, the following holds:

$$\Pr\left[\begin{array}{l|l} D = \mathbf{Dec}(prm, \mathbf{Enc}(conf, \mathbf{AONT}(& (prm, msk) \leftarrow \mathbf{Setup} \\ prm, conf, D), pk^{id_{du}}), sk^{id_{du}}) & (sk^{id_{du}}, pk^{id_{du}}) \leftarrow \mathbf{KG} \\ & conf \leftarrow \mathbf{IdxG} \end{array}\right] = 1$$

C. Threat Model and Security Requirements

In this section, we first present the trust evaluation assumptions for each entity's trust and reliability, then we explain the main threat model in our P^2HDS scheme.

1) Trust Assumptions: In the proposed system, we assume the following about each entity:

- The CA is assumed to be a trusted party in the sense that it correctly generates the required keys for the DOs and DUs and applies the required attribute-based access control policies. The CA is expected to cooperate with honest parties in fulfilling these tasks, but it is assumed never to collude with adversarial entities or act maliciously against honest parties.
- The ESs are operated by external service providers, not controlled by DOs or the CA. They are therefore deemed untrustworthy. They pose risks such as unauthorized access or attempts to derive sensitive information from the encrypted data stored within. An ES might try to learn about encrypted blocks for curiosity, or with malicious intentions to achieve competitive intelligence or sell information illegally to other entities.
- The DOs are presumed to process their data properly and safeguard their keys without sharing them, but we also consider that a DO might act maliciously to access other DOs' data.
- The DUs are potentially malicious individuals, who can collude among themselves or with ESs to gain access to outsourced data illegally. For instance, a DU lacking proper authorization (i.e., whose attributes do not meet the access policy) can try to access data by cooperating with ESs or DUs.

2) Threat Model: According to the above trust assumptions, the main security threat in our system is unauthorized data access. An adversary, either an unauthorized DU or a coordinated coalition of unauthorized DUs and compromised ESs, attempts to gain access to encrypted data stored on ESs without proper authorization, bypassing a DO's established access policies and cryptographic protections.

To address this threat, only authorized DUs whose attributes meet the required access policies are allowed access to the data, and encrypted files must not reveal any information about their data. To ensure this requirement, the P^2HDS scheme must achieve IND-CPA security. This strong confidentiality guarantee ensures that even if an adversary observes multiple ciphertexts and has some control over the plaintexts being encrypted, they cannot distinguish between encryptions of two chosen messages with more than a negligible advantage. This directly prevents unauthorized parties, including those involved in collusion, from obtaining any meaningful information about the sensitive data. We formally define IND-CPA security of P^2HDS in Section VI-A.

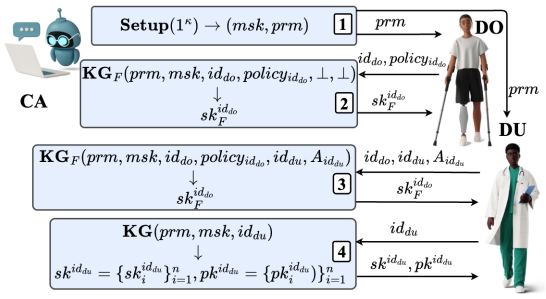


Figure 1: Phase 1, Initialization.

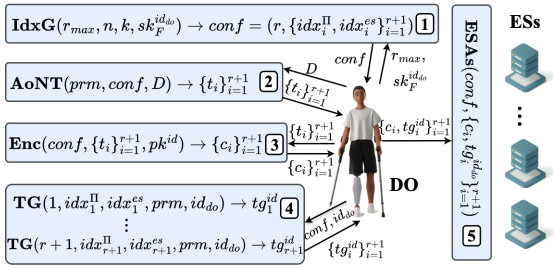


Figure 2: Phase 2, Data outsourcing.

IV. The P^2HDS approach

In this section, we describe our designed P^2HDS construction in detail and demonstrates its application in a decentralized storage system.

As mentioned in Section III-A, P^2HDS consists of three main phases: Initialization, Data outsourcing, and Data retrieval, each of which is thoroughly detailed below (Figures 1, 2, and 3). Figure 4 provides a detailed description of all the algorithms that form our proposed P^2HDS scheme and used in each of the following phases.

1) Initialization: At this stage, CA, guided by its expectations regarding the system security and efficiency, determines a security parameter κ and considers a set of encryption schemes $\{\Pi_i\}_{i=1}^n$. Then, it runs the Setup(1^κ) algorithm to initiate each of these schemes to produce their required parameters. Then, it shares the system public parameters prm openly and keeps msk securely. Afterward, it generates a configuration key $sk_F^{id_{do}}$ for each DO with the identifier id_{do} and access policy $policy_{id_{do}}$ by running $\mathrm{KG}_F(prm, msk, id_{do}, policy_{id_{do}}, \perp, \perp)$. For each DU also the configuration key is generated by running $\mathrm{KG}_F(prm, msk, id_{do}, policy_{id_{do}}, id_{du}, A_{id_{du}})$, where id_{du} represents the DU's identifier and $A_{id_{du}}$ is the associated attribute set (e.g., organizational role, expertise level, department, or geographical region). Here, id_{do} is the identifier of its desired DO (see Figure 1).

2) Data outsourcing: When a DO, identified by id_{do}, wishes to share a data set D with DU with the identifier id_{du}, it employs our hybrid encryption method to ensure both the confidentiality and efficiency of the data set. The first requirement of this approach is to divide the data

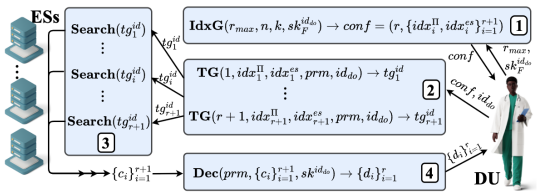


Figure 3: Phase 3, Data retrieval.

set into several blocks and determine a configuration for the encryption scheme and storage method of each block. Additionally, for security purposes, this configuration should be obfuscated so that the encryption methods are indistinguishable. To this end, the DO executes the index generation algorithm IdxG($r_{max}, n, k, sk_F^{id_{do}}$) using the config key $sk_F^{id_{do}}$ to generate a data set outsourcing configuration $conf = (r, \{idx_i^{\Pi}, idx_i^{es}\}_{i=1}^{r+1})$ as the number of blocks plus the indices of the encryption plans and the indices of ESs for each data block. In this algorithm, r_{max} is a threshold for the number of blocks, n is the number of encryption schemes, and k is the number of ESs.

Now, it applies an AONT transformation to a data set D by performing the AONT($prm, conf, D$) algorithm and outputs $r+1$ transferred data block $\{t_i\}_{i=1}^{r+1}$. This will make the security of the P^2HDS scheme as secure as the securest one, depending on the security of the AONT transformation. Afterward, the DO runs Enc($conf, \{t_i\}_{i=1}^{r+1}, pk^{id_{du}}$) to encrypt transferred blocks in $\{t_i\}_{i=1}^{r+1}$ to $\{c_i\}_{i=1}^{r+1}$ according to the encryption schemes' indices $\{idx_i^{\Pi}\}_{i=1}^{r+1}$ in $conf$. Then, to ensure that data blocks are retrievable for DUs, the DO will tag each block using the configuration $conf$. This tagging process will ensure that the tag contains the specific configuration of the data block without disclosing any details about the total number of blocks, the encryption method used, or the index of the ES. To achieve this, the DO will run the algorithm TG($prm, a, idx_a^{\Pi}, idx_a^{es}, sk_F^{id_{do}}$) to generate tag $tg_i^{id_{do}}$ for the encrypted block c_i. Finally, it executes the algorithm ESAs($conf, \{c_i, tg_i^{id}\}_{i=1}^{r+1}$) to outsource each pair (c_i, tg_i^{id}) to the corresponding ES indexed idx_i^{es} in $conf$ (see Figure 2).

3) Data retrieval: This is the last step, where the DU with identifier id_{du}, along with the ESs indexed in $conf$, will perform it. First, the DU performs again the index generation algorithm IdxG($r_{max}, n, k, sk_F^{id_{do}}$) using the config key $sk_F^{id_{do}}$ which is same as above to find the $conf$. Now, it needs to generate the tag associated with each index in the $conf$. To this end, the DU will run the algorithm TG($prm, a, idx_a^{\Pi}, idx_a^{es}, sk_F^{id}$) to ask from ES indexed idx_a^{es} for the date tagged $tg_a^{id_{do}}$. Then, the es_a receiving the query $tg_a^{id_{do}}$, search for the associated data block by running Search($tg_a^{id_{do}}$), and returns a ciphered data block c_a or $\perp$, there were no data match with $tg_a^{id_{do}}$. Finally, after collecting all ciphered blocks $\{c_i\}_{i=1}^{r+1}$, the DO executes the decryption algorithm Dec($prm, \{c_i\}_{i=1}^{r+1}, sk^{id}$) to retrieve the data set $D = d_1||\cdots||d_r$ (as shown in Figure 3).

V. Game-Theoretic Analysis of Configuration Selection

We consider the strategic interaction between a DO and an Attacker (ATT) in the configuration selection process of the P^2HDS scheme. The DO aims to select configurations in a way that minimizes the probability of being correctly guessed, while the ATT seeks to maximize its success probability by guessing the chosen configuration.

A. Game Definition

We define a two-player strategic game $\Gamma = (N, S, U)$ as follows. We consider the worst case that, once the ciphertexts are observed, the encryption scheme used for each block can be identified by the attacker. Therefore, the strategic choice of the DO is reduced to the number of blocks and their assignment to ESs.

- $N = \{do, att\}$ is the set of players, comprising a DO and an ATT.
- $S = S_{do} \times S_{att}$ is the strategy space. The DO's strategy space is $S_{do} = \{conf_1, conf_2, \ldots, conf_m\}$, the finite set of possible configurations. Each configuration specifies the number of blocks $r \in \{1, \ldots, r_{\max}\}$, and for each $i = 1, \ldots, r+1$, an ES index $idx^i \in \{1, \ldots, k\}$. The total number of configurations is
$$m = r_{\max} \cdot k^r,$$
where $r_{\max}$ is the maximum number of blocks. The ATT's strategy space is $S_{att} = \{guess_1, guess_2, \ldots, guess_m\}$ where each $guess_j$ corresponds to one possible configuration in S_{do}.
- $U = \{U_{do}, U_{att}\}$ is the set payoff functions, where:
  - $U_{do}(s_{do}, s_{att}) = \mathbb{I}(s_{do}, s_{att}) - 1$ is the DO's payoff,
  - $U_{att}(s_{do}, s_{att}) = 1 - \mathbb{I}(s_{do}, s_{att})$ is the ATT's payoff.

  Here, the indicator function $\mathbb{I}(\cdot)$ is equal to 1 if $s_{do} \neq s_{att}$ and 0 if $s_{do} = s_{att}$, which ensures a zero-sum game. The condition $s_{do} = s_{att}$ denotes that the ATT's chosen strategy matches the strategy selected by the DO, meaning the ATT has correctly guessed the selected configuration of DO.

B. Nash Equilibrium (NE) Analysis

We analyze pure and mixed NE strategy profile in this section.

1) Pure NE: A pure NE strategy profile requires $(s_{do}^*, s_{att}^*) \in S_{do} \times S_{att}$ such that no player benefits from unilateral deviation. For any fixed $s_{do}^* = conf_i$ selected by the DO for a $i \in \{1, \ldots, m\}$, the ATT can choose $s_{att}^* = guess_j$ for a $j \in \{1, \ldots, m\}$, resulting in $U_{do} = -1$, $U_{att} = 1$. If the DO deviates to $conf_k \neq conf_j$, $U_{do}(conf_k, guess_j) = 0$, improving from -1, the ATT can always match again, so there is no stable fixed point and accordingly no pure NE strategy profile equilibrium.

2) Mixed NE: In a mixed NE, the DO randomizes over S_{do} with $p = (p_1, \ldots, p_m)$, $\sum p_i = 1$, and ATT with $q = (q_1, \ldots, q_m)$, $\sum q_i = 1$. So, the expected payoffs for each player will be:
$$E[U_{do}] = -\sum_{i=1}^{m} p_i q_i, \quad E[U_{att}] = \sum_{i=1}^{m} p_i q_i.$$

1) Setup(1^κ) $\rightarrow (msk, prm)$: This algorithm takes as input a security parameter κ. First, for a natural number n, it picks n encryption scheme $\{\Pi_i\}_{i=1}^n$. For each $i \in \{1, \ldots, n\}$, $\Pi_i = (\text{Setup}_i, \text{KGen}_i, \text{Enc}_i, \text{Dec}_i)$. Moreover, it selects pseudo-random functions (PRF) $F_1 : \{0,1\}^\kappa \times \{0,1\}^{\log(r+1)} \rightarrow \{0,1\}^\kappa$, $F_2 : \{0,1\}^\ell \times \{0,1\}^{\log(r)} \rightarrow \{0,1\}^\ell$, $F_3 : \{0,1\}^\kappa \times \{0,1\}^* \rightarrow \{0,1\}^\kappa$, and $F_4 : \{0,1\}^\kappa \times \{0,1\}^* \rightarrow \{0,1\}^{\log(r_{max}+1)}$, along with and hash functions $H_1 : (\{0,1\}^\ell)^r \rightarrow \{0,1\}^\ell$ and $H_2 : \{0,1\}^{\log(r)} \times \{0,1\}^\kappa \times \{0,1\}^\kappa \times \{0,1\}^\kappa \rightarrow \{0,1\}^*$, where ℓ is a polynomial function of the security parameter κ: $\ell = p(\kappa)$. By running $\text{Setup}_i(1^\kappa)$ for each $i \in \{1, \ldots, n\}$, it generates (prm_i, msk_i), where prm_i and msk_i are public parameters and master secret key associated with Π_i. Afterward, it selects a random value $s \in \mathbb{Z}$. Finally it outputs $msk = (\kappa, s, \{msk_i\}_{i=1}^n$ and $prm = (\kappa, n, F_1, F_2, F_3, F_4, H_1, \{\Pi_i\}_{i=1}^n, \{prm_i^o\}_{i=1}^n)$.

2) KG$(prm, msk, id) \rightarrow (sk^{id}, pk^{id})$: Given public parameters prm, master secret key msk, an identifier id, it runs $\text{KGen}_i(1^n, prm_i, msk_i, id)$ for each $i \in \{1, \ldots, n\}$ and generates key pair (sk_i^{id}, pk_i^{id}) associated with identifiers id, where pk_i^{id} and sk_i^{id} are public key and secret key associated with Π_i. It outputs the secret key set $sk^{id} = \{sk_i^{id}\}_{i=1}^n$ and the public key set $pk^{id} = \{pk_i^{id}\}_{i=1}^n$.

3) $\text{KG}_F(prm, msk, id, policy_{id}, id', A_{id'}) \rightarrow (sk_F^{id})$: Given public parameters prm, master secret key msk, an identifier id and its associated policies $policy_{id}$, the following process occurs:

   - If both id' and $A_{id'}$ are $\perp$, the algorithm interprets this as the owner case and returns $sk_F^{id} = F_3(msk, id \parallel policy_{id})$.
   - If $id' \neq \perp$ and $A_{id'} \neq \perp$, it checks whether $A_{id'}$ satisfies $policy_{id}$. If so, it returns $sk_F^{id} = F_3(msk, id \parallel policy_{id})$; otherwise, it returns $sk_F^{id} = \perp$.

4) IdxG$(r_{max}, n, k, sk_F^{id}) \rightarrow conf = (r, \{idx_i^\Pi, idx_i^{es}\}_{i=1}^{r+1})$: Given natural numbers r_{max}, n, k, this algorithm first sets $r = (F_4(sk_F^{id}, n||k||r_{max}) \mod (r_{max})) + 1$. Then it computes $idx_i^\Pi = F_1(sk_F^{id}, i||0) \mod (n+1)$ and $idx_i^{es} = F_1(sk_F^{id}, i||1) \mod (k+1)$ for $i = \{1, \ldots, r+1\}$. It outputs the data outsourcing configuration $conf = (r, \{idx_i^\Pi, idx_i^{es}\}_{i=1}^{r+1})$.

5) AONT$(prm, conf, D) \rightarrow (\{t_i\}_{i=1}^{r+1})$: This algorithm divides the data set D into r blocks of length l, $d_1 || \ldots || d_r$. Then, it selects $\alpha \leftarrow \{0,1\}^\ell$, and for $i = \{1, \ldots, r\}$ it computes $t_i = d_i \oplus F_2(\alpha, i)$. Finally, it calculates $t_{r+1} = \alpha \oplus H_1(t_1, \ldots, t_r)$ and returns $\{t_i\}_{i=1}^{r+1}$ (This design is driven by the AONT scheme proposed in [36]).

6) Enc$(conf, \{t_i\}_{i=1}^{r+1}, pk^{id}) \rightarrow (\{c_i\}_{i=1}^{r+1})$: This algorithm runs $enc_i(t_i, pk_i^{id})$ of Π_i indexed idx_i^Π for each $i \in \{1, \ldots, r+1\}$ to encrypt each transferred block t_i to ciphertext c_i. It outputs $\{c_i\}_{i=1}^{r+1}$.

7) TG$(prm, a, idx_a^\Pi, idx_a^{es}, sk_F^{id}) \rightarrow (tg_a^{id})$: This algorithm takes an index $a \in \{1, \ldots, r+1\}$, along with its corresponding encryption scheme and ES indices, idx_a^Π and idx_a^{es} from $conf = (r, \{idx_i^\Pi, idx_i^{es}\}_{i=1}^{r+1})$ plus the secret key sk_F^{id}, then returns a tag $tg_a^{id} = H_2(a||idx_a^\Pi||idx_a^{es}||sk_F^{id})$.

8) ESAs$(conf, \{t_i, tg_i^{id}\}_{i=1}^{r+1}) \rightarrow (\{status_i\}_{i=1}^{r+1})$: For each $i = \{1, \ldots, r+1\}$, it attempts to assign the data pair (t_i, tg_i^{id}) to the corresponding ES indexed idx_i^{es}. If the assignment succeeds, it sets $status_i$success; otherwise, $status_i =$ failure). Finally it returns $\{status_i)\}_{i=1}^{r+1}$.

9) Search$(tg^{id}) \rightarrow (c_i$ or $\perp)$: On input a tag tg^{id}, this algorithm will search for a ciphered block that is tagged tg^{id}. If it finds, it will return c_i associated with tg^{id}. Otherwise, it returns $\perp$.

10) Dec$(prm, \{c_i\}_{i=1}^{r+1}, sk^{id}) \rightarrow D$: It first retrieves the transferred blocks $\{t_i\}_{i=1}^{r+1}$ by running $\text{Dec}_i(c_i, sk_i^{id})$ of Π_i indexed idx_i^Π for each $i \in \{1, \ldots, r+1\}$ to decrypt each ciphertext c_i. Then it sets $\alpha = t_{r+1} \oplus H_1(t_1, \ldots, t_r)$ and recover $d_i = t_i \oplus F_2(\alpha, i)$ for $i = \{1, \ldots, r\}$. It outputs $D = d_1 || \cdots || d_r$.

Figure 4: The concrete construction of our FGDM.

ATT maximizes $E[U_{att}]$ by setting $q_j = 1$ if $p_j = \max_i p_i$, achieving $E[U_{att}] = \max_i p_i$, $E[U_{do}] = -\max_i p_i$. The DO maximizes $E[U_{do}]$ by minimizing $\max_i p_i$, at $p_i = 1/m$, resulting in:

$$E[U_{att}] = \frac{1}{m}, \quad E[U_{do}] = -\frac{1}{m}$$

This is a mixed NE strategy profile, as deviations are countered.

C. Configuration Selection Analysis

Theorem 1. The DO's best response is to select the configuration randomly with uniform probability $1/m$, where $m = r_{\max} \cdot k^r$.

Since the game Γ is zero-sum and has a diagonal game matrix $A = diag(a_{11} = 1, \ldots, a_{mm} = 1)$, as demonstrated by Rass et al. in section 12.3.3 of [43], we have a closed form expression for the saddle-point

value as $\left(\sum_{i=1}^{m} \frac{1}{a_{ii}}\right)^{-1} = (m \cdot 1)^{-1}$, with the same mixed equilibrium strategy for both players, $(p_i)_{i=1}^{m} = (q_i)_{i=1}^{m} = \left(\frac{1}{m}\right)_{i=1}^{m}$.

The P^2HDS scheme achieves the optimal mixed strategy in practice by using a PRF scheme in the IdxG algorithm to generate the configuration parameters, including the number of blocks r and the ES indices $\{idx_i^{es}\}_{i=1}^{r+1}$. This construction ensures a uniform distribution over the $m = r_{\max} \cdot k^r$ possible configurations, which aligns with the equilibrium strategy where the DO randomizes with probability $1/m$ to minimize the attacker's success probability to $E[U_{do}] = -1/m$. By standard indistinguishability guarantees, the output of the PRF is computationally indistinguishable from truly uniform sampling for any PPT adversary. Thus, without explicitly enumerating all m configurations, which would be exponential in r, the scheme efficiently and naturally implements the game-theoretically optimal defense.

VI. Security Analysis

In this section, we provide the formal security definition for our P^2HDS scheme, followed by its security proof.

A. Adversarial Model

The primary focus of P^2HDS scheme is ensuring confidentiality, so we specifically define the security requirement to achieve confidentiality within the context of our scheme.

According to the Section III-C, ESs and unauthorized DUs are not trusted as they may collude to violate data confidentiality. When a DO encrypts the data set D using the system defined in Section III-B for a DU identified by id_{du}^*, some unauthorized DUs may collude with compromised ESs to gain access to information about the data. Naturally, the DU with identifier id_{du}^*, who is trusted by the DO, does not participate in this collusion, as they possess the necessary keys to legitimately access the data and thus have no reason and incentive to share such information.

To represent this threat as a single adversary, we define an adversary $\mathcal{A}$ that captures unauthorized DUs conspiring with compromised ESs. In the worst case, we assume that all DUs except the authorized one with identifier id_{du}^* may collude with the ESs, i.e., among N DUs, at most $N-1$ can be compromised.

The purpose of $\mathcal{A}$ is to reveal information about the encrypted data that would allow it to distinguish between possible original plaintexts. Specifically, $\mathcal{A}$ aims to distinguish between two chosen data sets D_0 and D_1, where one has been processed and encrypted, thereby inferring information about the actual content of the data. Suppose a stateful PPT challenger $\mathcal{C}$ and the P^2HDS scheme Π with the algorithms in Section III-B. In what follows, we simulate such an adversary in the game described.

Indistinguishability Experiment: We define the experiment through the following steps:

- Step 1: The challenger $\mathcal{C}$ considers initially empty lists L_{id}^{du} and L_{id}^{do}, and starts setting up the experiment by first selecting a security parameter κ and executing $(prm, msk) \leftarrow Setup(1^\kappa)$. The resulting public parameters prm are then given to $\mathcal{A}$. Next, $\mathcal{A}$ designates a target DU identifier id_{du}^*, which will serve as the challenge identity throughout the game.
- Step 2: $\mathcal{A}$ asks polynomially many (in κ) queries to the oracles $\mathcal{O}_{\mathrm{KG}}(id_{du})$ and $\mathcal{O}_{\mathrm{KG}_F}(id_{do}, policy_{do}, id_{du}, A_{id_{du}})$ with the restriction that id_{du}^* cannot be used as input to either oracle, and $\mathcal{C}$ answers them as follows:
  - $\mathcal{O}_{\mathrm{KG}}(id_{du})$: On input $id_{du} \neq id_{du}^*$, $\mathcal{C}$ adds id_{du} to L_{id}^{du}, executes $(sk^{id_{du}}, pk^{id_{du}}) \leftarrow \mathrm{KG}(prm, msk, id_{du})$ and returns $(sk^{id_{du}}, pk^{id_{du}})$ to $\mathcal{A}$.
  - $\mathcal{O}_{\mathrm{KG}_F}(id_{do}, policy_{id_{do}}, id_{du}, A_{id_{du}})$: Given id_{do} and $id_{du} \neq id_{du}^*$, $\mathcal{C}$ first adds id_{do} and id_{du} to L_{id}^{do} and L_{id}^{du}, respectively, if $id_{du} \notin L_{id}^{du}$ and $id_{do} \notin L_{id}^{do}$. Then, it executes $sk_F^{id_{do}} \leftarrow KG_F(prm, msk, id_{do}, policy_{id_{do}}, id_{du}, A_{id_{du}})$ and returns $sk_F^{id_{do}}$ to $\mathcal{A}$.
- Step 3: When Step 2 is finished, $\mathcal{A}$ sends again the challenge identifier id_{du}^* along with a DO identifier id_{do}^* to $\mathcal{C}$. Then, $\mathcal{C}$ runs $(sk^{id_{du}^*}, pk^{id_{du}^*}) \leftarrow \mathrm{KG}(prm, msk, id_{du}^*)$, and $sk_F^{id_{do}^*} \leftarrow KG_F(prm, msk, id_{do}^*, policy_{id_{do}^*}, id_{du}^*, A_{id_{du}^*})$ and returns $pk^{id_{du}^*}$ to $\mathcal{A}$. Then, $\mathcal{A}$ makes a polynomial number of queries (in κ) to the oracles $\mathcal{O}_{\mathrm{KG}}(id_{du})$ and $\mathcal{O}_{\mathrm{KG}_F}(id_{do}, policy_{do}, id_{du}, A_{id_{du}})$. If $id_{du} \neq id_{du}^*$, $\mathcal{C}$ answers them as Step 2, otherwise, it does not respond to queries.
- Step 4 (Challenge): The adversary $\mathcal{A}$ selects two data sets D_0 and D_1 of equal length and submits to $\mathcal{C}$. The challenger $\mathcal{C}$ selects a bit $b \in \{0,1\}$ and generates the challenge ciphertext as follows:
  1) Index Generation:
     - Compute $r = (F_4(sk_F^{id_{do}^*}, n||k||r_{max}) \bmod r_{max}) + 1$.
     - For $i = 1, \ldots, r+1$, set $idx_i^{\Pi} = F_1(sk_F^{id_{do}^*}, i||0) \bmod (n+1)$ and $idx_i^{es} = F_1(sk_F^{id_{do}^*}, i||1) \bmod (k+1)$.
     - Output: $conf = (r, \{idx_i^{\Pi}, idx_i^{es}\}_{i=1}^{r+1})$.
  2) AONT Transform:
     - Split D_b into r blocks $\{d_i\}_{i=1}^{r}$, and select $\alpha \xleftarrow{\$} \{0,1\}^{\ell}$.
     - For $i = 1, \ldots, r$, set $t_i = d_i \oplus F_2(\alpha, i)$, and compute $t_{r+1} = \alpha \oplus H_1(t_1, \ldots, t_r)$.
     - Output: $\{t_i\}_{i=1}^{r+1}$.
  3) Encryption:
     - For $i \in \{1, \ldots, r+1\}$ compute $c_i = \mathrm{Enc}_i(t_i, pk_i^{id_{du}^*})$.
     - Output: $\{c_i\}_{i=1}^{r+1}$.
  4) Tag Generation:
     - For $i \in \{1, \ldots, r+1\}$ compute

$tg_i^{id_o^*} = H_2(i||idx_i^{\Pi}||idx_i^{es}||sk_F^{id_{do}^*})$.

– Output: $\{tg_i^{id_o^*}\}_{i=1}^{r+1}$.

Finally $\mathcal{C}$ returns $\{tg_i^{id_o^*}\}_{i=1}^{r+1}$ to $\mathcal{A}$.

- Step 5 (Guess): $\mathcal{A}$ returns a guess $b' \in \{0,1\}$.

The winner of the IND-CPA game is $\mathcal{A}$ if and only if $b' = b$. We define the advantage of $\mathcal{A}$ against a P^2HDS scheme denoted as Π using the following function:

$$\mathrm{Adv}_{\mathcal{A}}^{\Pi}(\kappa) = |Pr(b = b') - 1/2|,$$

where κ is the security parameter, and $\mathcal{A}$ is the PPT adversary in the described indistinguishability experiment.

Definition 2. A P^2HDS scheme, Π, is secure against IND-CPA if $\mathrm{Adv}_{\mathcal{A}}^{\Pi}(\kappa)$ is a negligible function in κ for any PPT adversary $\mathcal{A}$.

B. Security Proof

Theorem 2. The P^2HDS scheme Π is IND-CPA secure in the random oracle model, assuming at least one underlying encryption schemes is IND-CPA secure. Specifically, for any PPT adversary $\mathcal{A}$, there exists a negligible function $\mathrm{negl}(\kappa)$ such that:

$$\mathrm{Adv}_{\mathcal{A}}^{\Pi}(\kappa) \leq \mathrm{negl}(\kappa)$$

Proof. Assume there exists a PPT adversary $\mathcal{A}$ that breaks the IND-CPA security of P^2HDS with non-negligible advantage, i.e., $\mathrm{Adv}_{\mathcal{A}}^{\Pi}(\kappa) \geq \epsilon(\kappa)$, where $\epsilon(\kappa)$ is non-negligible. Let $j \in \{1, \ldots, r+1\}$ be the least index such that Π_j is IND-CPA secure. We construct a distinguisher $\mathcal{D}$ that runs $\mathcal{A}$ as a subroutine to break the IND-CPA security of Π_j, aiming to distinguish whether a ciphertext encrypts a chosen message using Π_j or is random.

Reduction: The distinguisher $\mathcal{D}$ receives a public key for Π_j from the IND-CPA challenger of Π_j, $\mathcal{C}_j$, and simulates the IND-CPA game for $\mathcal{A}$. It generates keys for all Π_i (for $i \neq j$) and sets up the P^2HDS parameters. When $\mathcal{A}$ submits messages D_0 and D_1, $\mathcal{D}$ picks a random bit b, splits D_b into blocks $t_1, \ldots, t_{r+1}$, and encrypts them. For blocks $i \neq j$, $\mathcal{D}$ encrypts normally using Π_i. For block j, $\mathcal{D}$ sends t_j to challenger $\mathcal{C}_j$, which returns c_j. $\mathcal{D}$ uses c_j, completes the encryption process, and gives $\mathcal{A}$ the tags, as described in the IND-CPA game. $\mathcal{A}$ retrieves the ciphertexts and makes more queries to the oracles in the IND-CPA game. $\mathcal{D}$ answers $\mathcal{A}$'s queries to H_1 as a random oracle as follows:

- For $H_1(t_1, \ldots, t_r)$, it returns a random ℓ-bit string and maintains it in a table to keep answers consistent.

Then, $\mathcal{A}$ outputs a guess b'. If $b' = b$, $\mathcal{D}$ outputs 1 (guessing c_j is real); otherwise, it outputs 0 (guessing c_j is random).

Advantage analysis: Now, We calculate $\mathcal{D}$'s distinguishing advantage to break Π_j in order to evaluate $\mathcal{A}$'s advantage to break Π in the IND-CPA experiment.

- Real Case (R_1): If c_j is the real encryption of t_j, $\mathcal{A}$ is playing the real game. The probability $\mathcal{A}$ guesses b correctly is $\Pr[b' = b]$, and $\mathcal{D}$ outputs 1 with the same probability, i.e. $\Pr[b' = b] = \frac{1}{2} + \mathrm{Adv}_{\mathcal{A}}^{\Pi}(\kappa)$.
- Random Case (R_2): If c_j is a random string (i.e., not a valid encryption), then it carries no information about t_j, regardless of the security of Π_j. Hence $\mathcal{A}$ cannot obtain t_j from c_j. To recover D_b, however, $\mathcal{A}$ needs all blocks to compute α, which requires knowing t_j. The probability of guessing t_j correctly is $2^{-\ell}$. With $q_{H_1} \leq p(\kappa)$ queries to H_1, the chance of guessing right is at most $q_{H_1} \cdot 2^{-\ell}$. If $\mathcal{A}$ guesses correctly, it recovers D_b and guesses b with probability 1; otherwise, it guesses randomly (probability $\frac{1}{2}$). So, the probability $\mathcal{A}$ guesses correctly is:

$$(q_{H_1} \cdot 2^{-\ell}) \cdot 1 + (1 - q_{H_1} \cdot 2^{-\ell}) \cdot \frac{1}{2} = \frac{1}{2} + \frac{q_{H_1} \cdot 2^{-\ell}}{2}.$$

$\mathcal{D}$'s advantage in its IND-CPA game is the difference in its output probabilities:

$$\mathrm{Adv}_{\mathcal{D}}^{\Pi_j}(\kappa) = |\Pr[D = 1 \text{ in } R_1] - \Pr[D = 1 \text{ in } R_2]| = |\Pr[b = b'] - (\frac{1}{2}$$

We know that $\mathrm{Adv}_{\mathcal{A}}^{\Pi}(\kappa) = |\Pr[b = b'] - \frac{1}{2}|$, so

$$\mathrm{Adv}_{\mathcal{D}}^{\Pi_j}(\kappa) = |\mathrm{Adv}_{\mathcal{A}}^{\Pi}(\kappa) - \frac{q_{H_1} \cdot 2^{-\ell}}{2}|.$$

Since $\mathrm{Adv}_{\mathcal{A}}^{\Pi}(\kappa) \geq \epsilon(\kappa)$, and assuming $\ell = \kappa$ (so $\frac{q_{H_1} \cdot 2^{-\ell}}{2} \leq \frac{p(\kappa)}{2^{\kappa+1}}$, which is negligible), we have:

$$\mathrm{Adv}_{\mathcal{D}}^{\Pi_j}(\kappa) \geq \epsilon(\kappa) - \mathrm{negl}(\kappa).$$

Since $\epsilon(\kappa)$ is non-negligible, $\mathrm{Adv}_{\mathcal{D}}^{\Pi_j}(\kappa)$ is non-negligible, contradicting the IND-CPA security of Π_j, which requires $\mathrm{Adv}_{\mathcal{D}}^{\Pi_j}(\kappa) \leq \mathrm{negl}(\kappa)$. Thus, no such $\mathcal{A}$ exists, and P^2HDS is IND-CPA-secure. □

VII. Performance Analysis

This section details our simulation results, evaluating the performance of our proposed approach. We start by describing the simulation setup, then move into the numerical results and their analysis.

Simulation Setup: To simulate the proposed approach, the system was set up using five encryption schemes ($n = 5$), 50 ESs ($k = 50$), and a maximum block count of 50 ($r_{max} = 50$). The five encryption schemes considered in this experiment are RSA [44], ECC [45], ElGamal [46], NTRU [47], and McEliece [48]. Specifically, RSA is a widely used public-key cryptosystem based on the hardness of integer factorization [44]; ECC (Elliptic Curve Cryptography) achieves similar levels of security with significantly smaller keys by relying on the elliptic curve discrete logarithm problem [45]; ElGamal provides semantic security through probabilistic encryption and is founded on the discrete logarithm problem [46]; NTRU is a lattice-based cryptosystem that offers high efficiency and resistance to quantum attacks, making it a strong candidate for post-quantum cryptography [47]; and McEliece is a code-based post-quantum cryptosystem that leverages error-correcting codes for quantum-resistant security, offering robust protection but requiring larger keys [48].

The experiments were conducted using Python on a Mac M1 platform, equipped with an Apple M1 chip and 16 GB RAM, running macOS 15.5, leveraging the sympy[3] library for symbolic mathematics, particularly for polynomial inversions in the NTRU algorithm, and the cryptography[4] library for implementing cryptographic primitives. The dataset utilized in the experiments was the Wearable Exam Stress Dataset from PhysioNet[5], comprising physiological signals. Each fixed-size configuration is tested over 100 random instances to ensure statistical reliability. Then, relevant reported metrics are evaluated as the average across these 100 instances.

Numerical Results: Figure 5a illustrates how encryption time varies across different schemes as the plaintext data size increases from 1 KB to 16 KB. The Z-axis shows the encryption time in seconds, and the trend for each scheme reflects how well it scales with larger inputs. Notably, P^2HDS exhibits a consistent and efficient performance trend across all data sizes. While some schemes show significant increases in encryption time as the data size grows, P^2HDS maintains a relatively modest and linear growth. For smaller input sizes (1–2 KB), its encryption latency closely aligns with the fastest schemes in the comparison. As the data size scales up to 16 KB, P^2HDS continues to demonstrate efficiency, staying well below the encryption times observed for more computationally expensive approaches like RSA and McEliece. The modest increase in P^2HDS's encryption time across the data range indicates a well-optimized hybrid integration of multiple encryption schemes. This shows that P^2HDS is not only practical for small-scale data but also scales effectively with increasing input sizes, making it suitable for a broad range of applications.

Figure 5b presents the ciphertext size generated by each scheme when encrypting data ranging from 1 KB to 16 KB. Ciphertext size is a critical performance metric, especially in bandwidth-constrained environments. As shown in this chart, although the ciphertext sizes of P^2HDS are naturally larger than those of some lightweight schemes due to its hybrid construction, they still remain smaller than the ciphertexts produced by high-overhead schemes such as ECC and ElGamal. As the plaintext data size increases, the ciphertext size under P^2HDS grows proportionally, but the slope of this growth is clearly more moderate than that of the most storage-intensive schemes. For the largest tested input of 16 KB, P^2HDS remains well within a ciphertext size of approximately 200 KB, illustrating a favorable trade-off between enhanced security and storage overhead. This positions P^2HDS as a viable option for use cases where ciphertext size must be minimized without sacrificing robustness.

[3]Python library for symbolic mathematics, available at https://www.sympy.org/

[4]Python library for secure communication, available at https://cryptography.io/

[5]https://physionet.org/content/wearable-exam-stress/1.0.0/

Figure 5c measures decryption time in seconds as a function of ciphertext size, ranging from 50 KB to 250 KB. The performance of P^2HDS on this metric underscores one of its major strengths: decryption efficiency. Despite its hybrid nature, P^2HDS consistently maintains a fast decryption process across all tested ciphertext sizes. Even for the largest ciphertexts, the decryption time for P^2HDS remains comfortably under 0.12 seconds. While traditional schemes like RSA suffer from significantly increasing decryption times, P^2HDS achieves predictable, low-latency decryption regardless of ciphertext growth. This makes it particularly well-suited for latency-sensitive applications in IoMT environments where swift data recovery is critical.

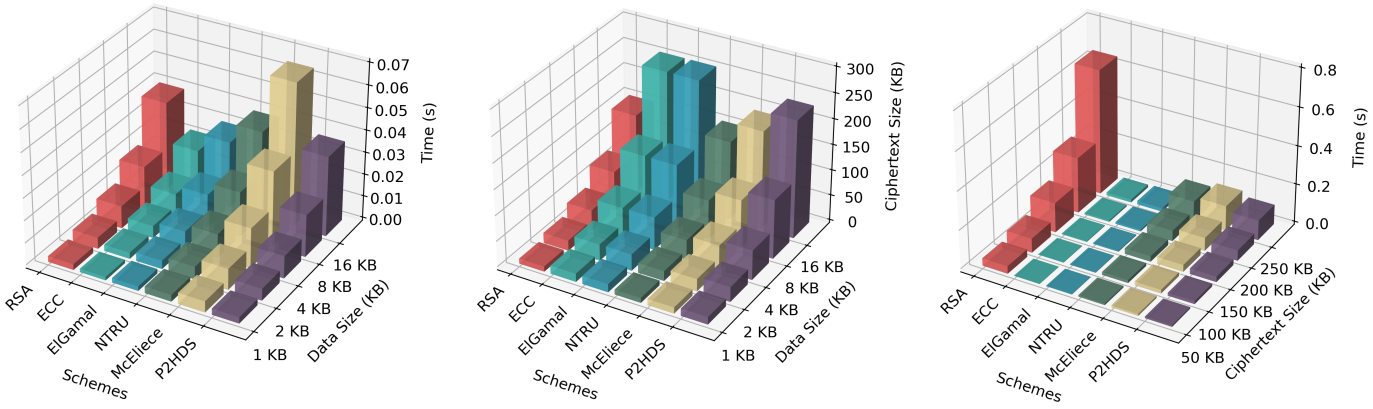
Overall, the performance results highlight that P^2HDS achieves a favorable balance between efficiency, scalability, and storage overhead across encryption, ciphertext size, and decryption metrics. While some classical schemes such as ECC and ElGamal may appear more lightweight in certain scenarios, their security does not extend to the post-quantum setting. In contrast, P^2HDS can be instantiated with post-quantum secure primitives, thereby combining competitive performance with long-term robustness. This makes P^2HDS not only efficient for current edge computing applications but also adaptable to future environments where quantum resistance becomes essential.

VIII. Conclusion

The Privacy-Preserving Hybrid Data Sharing (P^2HDS) scheme addresses the critical need for secure and efficient data sharing in the Internet of Medical Things (IoMT) within edge computing environments. By integrating multiple encryption schemes, P^2HDS achieves a robust balance between security and computational efficiency, making it well-suited for resource-constrained IoMT devices. Our formalization of P^2HDS as a generalized framework provides a flexible approach to secure data outsourcing, while our game-theoretic analysis optimizes the selection of encryption strategies to counter adversarial threats. The rigorous security proof establishes that P^2HDS inherits the IND-CPA security of its most secure constituent scheme in the random oracle model, ensuring strong protection against privacy breaches. Experimental results further confirm that P^2HDS delivers performance close to the average of its constituent schemes, significantly reducing resource demands compared to computationally intensive protocols. These advancements make P^2HDS a promising solution for privacy-preserving applications, such as secure machine learning in healthcare. Future work will explore extensions to dynamic datasets and the integration of P^2HDS with federated learning frameworks to further enhance privacy and scalability in IoMT systems.

References

[1] G. J. Joyia, R. M. Liaqat, A. Farooq, and S. Rehman, "Internet of medical things (iomt): Applications, benefits and future challenges in healthcare domain." J. Commun., vol. 12, no. 4, pp. 240–247, 2017.



(a) Average encryption time (b) Average storage overhead (c) Average decryption time

Figure 5: Performance Evaluation of the P^2HDS scheme.

[2] Z. Ning, P. Dong, X. Wang, X. Hu, J. Liu, L. Guo, B. Hu, R. Y. Kwok, and V. C. Leung, "Partial computation offloading and adaptive task scheduling for 5g-enabled vehicular networks," IEEE Transactions on Mobile Computing, vol. 21, no. 4, pp. 1319–1333, 2020.

[3] X. Wang, Z. Ning, and S. Guo, "Multi-agent imitation learning for pervasive edge computing: A decentralized computation offloading algorithm," IEEE Transactions on Parallel and Distributed Systems, vol. 32, no. 2, pp. 411–425, 2020.

[4] M. Satyanarayanan, "The emergence of edge computing," Computer, vol. 50, no. 1, pp. 30–39, 2017.

[5] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," Future Generation Computer Systems, vol. 97, pp. 219–235, 2019.

[6] P. Porambage, J. Okwuibe, M. Liyanage, M. Ylianttila, and T. Taleb, "Survey on multi-access edge computing for internet of things realization," IEEE Communications Surveys & Tutorials, vol. 20, no. 4, pp. 2961–2991, 2018.

[7] B. S. Egala, A. K. Pradhan, V. Badarla, and S. P. Mohanty, "Fortified-chain: a blockchain-based framework for security and privacy-assured internet of medical things with effective access control," IEEE Internet of Things Journal, vol. 8, no. 14, pp. 11 717–11 731, 2021.

[8] G. Wu, S. Wang, Z. Ning et al., "Blockchain-enabled privacy-preserving access control for data publishing and sharing in the internet of medical things," IEEE Internet of Things Journal, vol. 9, no. 11, pp. 8091–8104, 2021.

[9] S. Rass and P. Schartner, "A unified framework for the analysis of availability, reliability and security, with applications to quantum networks," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 41, no. 1, pp. 107–119, 2010.

[10] ——, "Information-leakage in hybrid randomized protocols." Proceedings of the International Conference on Security and Cryptography, 2011, pp. 134–143.

[11] C. E. Shannon, "Communication theory of secrecy systems," The Bell system technical journal, vol. 28, no. 4, pp. 656–715, 1949.

[12] S. Even and O. Goldreich, "On the power of cascade ciphers," ACM Transactions on Computer Systems (TOCS), vol. 3, no. 2, pp. 108–116, 1985.

[13] U. M. Maurer and J. L. Massey, "Cascade ciphers: The importance of being first," Journal of Cryptology, vol. 6, no. 1, pp. 55–61, 1993.

[14] W. Aiello, M. Bellare, G. Di Crescenzo, and R. Venkatesan, "Security amplification by composition: The case of doubly-iterated, ideal ciphers," in Annual International Cryptology Conference. Springer, 1998, pp. 390–407.

[15] R. C. Merkle and M. E. Hellman, "On the security of multiple encryption," Communications of the ACM, vol. 24, no. 7, pp. 465–467, 1981.

[16] T. Soroceanu, N. Buchmann, and M. Margraf, "On multiple encryption for public-key cryptography," Cryptography, vol. 7, no. 4, p. 49, 2023.

[17] A. A. Giron, R. Custódio, and F. Rodríguez-Henríquez, "Post-quantum hybrid key exchange: a systematic mapping study," Journal of Cryptographic Engineering, vol. 13, no. 1, pp. 71–88, 2023.

[18] F. Giacon, F. Heuer, and B. Poettering, "Kem combiners," in IACR International Workshop on Public Key Cryptography. Springer, 2018, pp. 190–218.

[19] N. Bindel, J. Brendel, M. Fischlin, B. Goncalves, and D. Stebila, "Hybrid key encapsulation mechanisms and authenticated key exchange," in International Conference on Post-Quantum Cryptography. Springer, 2019, pp. 206–226.

[20] L. Huguenin-Dumittan and S. Vaudenay, "Fo-like combiners and hybrid post-quantum cryptography," in International Conference on Cryptology and Network Security. Springer, 2021, pp. 225–244.

[21] C. A. Asmuth and G. R. Blakley, "An efficient algorithm for constructing a cryptosystem which is harder

to break than two other cryptosystems," Computers & Mathematics with Applications, vol. 7, no. 6, pp. 447–450, 1981.
[22] A. Herzberg, "Folklore, practice and theory of robust combiners," Journal of Computer Security, vol. 17, no. 2, pp. 159–189, 2009.
[23] R. Zhang, G. Hanaoka, J. Shikata, and H. Imai, "On the security of multiple encryption or cca-security+ cca-security= cca-security?" in International Workshop on Public Key Cryptography. Springer, 2004, pp. 360–374.
[24] Y. Dodis and J. Katz, "Chosen-ciphertext security of multiple encryption," in Theory of Cryptography Conference. Springer, 2005, pp. 188–209.
[25] A. Fujioka, Y. Okamoto, and T. Saito, "Security of sequential multiple encryption," in International Conference on Cryptology and Information Security in Latin America. Springer, 2010, pp. 20–39.
[26] S. Hohenberger, A. Lewko, and B. Waters, "Detecting dangerous queries: A new approach for chosen ciphertext security," in Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, 2012, pp. 663–681.
[27] B. Goncalves and A. Mashatan, "Tightly secure pke combiner in the quantum random oracle model," Cryptography, vol. 6, no. 2, p. 15, 2022.
[28] M. Braithwaite, Experimenting with Post-Quantum Cryptography.
[29] E. Alkim, L. Ducas, T. Pöppelmann, and P. Schwabe, "Post-quantum key {Exchange—A} new hope," in 25th USENIX security symposium (USENIX Security 16), 2016, pp. 327–343.
[30] A. Hülsing, J. Rijneveld, J. Schanck, and P. Schwabe, "High-speed key encapsulation from ntru," in International Conference on Cryptographic Hardware and Embedded Systems. Springer, 2017, pp. 232–252.
[31] S. S. Stefan Kölbl, Rafael Misoczki, Securing tomorrow today: Why Google now protects its internal communications from quantum threats.
[32] L. V. Kris Kwiatkowski, The TLS Post-Quantum Experiment.
[33] C. P. Karen Easterbrook, Post-Quantum TLS.
[34] M. Anastasova, P. Kampanakis, and J. Massimo, "Pq-hpke: post-quantum hybrid public key encryption," Cryptology ePrint Archive, 2022.
[35] R. L. Rivest, "All-or-nothing encryption and the package transform," in International Workshop on Fast Software Encryption. Springer, 1997, pp. 210–218.
[36] V. Boyko, "On the security properties of oaep as an all-or-nothing transform," in Annual international cryptology conference. Springer, 1999, pp. 503–518.
[37] A. Desai, "The security of all-or-nothing encryption: Protecting against exhaustive key search," in Annual International Cryptology Conference. Springer, 2000, pp. 359–375.
[38] G. O. Karame, C. Soriente, K. Lichota, and S. Capkun, "Securing cloud data under key exposure," IEEE Transactions on Cloud Computing, vol. 7, no. 3, pp. 838–849, 2017.
[39] E. Bacis, S. De Capitani di Vimercati, S. Foresti, S. Paraboschi, M. Rosa, and P. Samarati, "Mix&slice: Efficient access revocation in the cloud," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, 2016, pp. 217–228.
[40] K. Kapusta and G. Memmi, "Enhancing data protection in a distributed storage environment using structure-wise fragmentation and dispersal of encrypted data," in 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE). IEEE, 2018, pp. 385–390.
[41] ——, "Circular aon: A very fast scheme to protect encrypted data against key exposure," in Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, 2018, pp. 2231–2233.
[42] Y. Gu, S. Akao, N. N. Esfahani, Y. Miao, and K. Sakurai, "On the security properties of combinatorial all-or-nothing transforms," in 2022 IEEE International Symposium on Information Theory (ISIT). IEEE, 2022, pp. 1447–1452.
[43] S. Rass, S. Schauer, S. König, and Q. Zhu, Cyber-Security in Critical Infrastructures: A Game-Theoretic Approach. SpringerNature, 2020. [Online]. Available: https://link.springer.com/book/10.1007/978-3-030-46908-5
[44] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120–126, 1978.
[45] V. S. Miller, "Use of elliptic curves in cryptography," in Conference on the theory and application of cryptographic techniques. Springer, 1985, pp. 417–426.
[46] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," IEEE transactions on information theory, vol. 31, no. 4, pp. 469–472, 1985.
[47] J. Hoffstein, J. Pipher, and J. H. Silverman, "Ntru: A ring-based public key cryptosystem," in International algorithmic number theory symposium. Springer, 1998, pp. 267–288.
[48] R. J. McEliece, "A public-key cryptosystem based on algebraic," Coding Thv, vol. 4244, no. 1978, pp. 114–116, 1978.