Efficient User-Centric Privacy-Friendly and Flexible Wearable Data Aggregation and Sharing

DOI:
10.1109/TCC.2024.3375801

Document Version
Accepted author manuscript

Efficient User-Centric Privacy-Friendly and Flexible Wearable Data Aggregation and Sharing

Khlood Jastaniah, Ning Zhang, and Mustafa A. Mustafa

***Abstract*—Wearable devices can offer services to individuals and the public. However, wearable data collected by cloud providers may pose privacy risks. To reduce these risks while maintaining full functionality, healthcare systems require solutions for privacy-friendly data processing and sharing that can accommodate three main use cases: (i) data owners requesting processing of their own data, and multiple data requesters requesting data processing of (ii) a single or (iii) multiple data owners. Existing work lacks data owner access control and does not efficiently support these cases, making them unsuitable for wearable devices. To address these limitations, we propose a novel, efficient, user-centric, privacy-friendly, and flexible data aggregation and sharing scheme, named SAMA. SAMA uses a multi-key partial homomorphic encryption scheme to allow flexibility in accommodating the aggregation of data originating from a single or multiple data owners while preserving privacy during the processing. It also uses ciphertext-policy attribute-based encryption scheme to support fine-grain sharing with multiple data requesters based on user-centric access control. Formal security analysis shows that SAMA supports data confidentiality and authorisation. SAMA has also been analysed in terms of computational and communication overheads. Our experimental results demonstrate that SAMA supports privacy-preserving flexible data aggregation more efficiently than the relevant state-of-the-art solutions.**

***Index Terms*—Wearables, Privacy, Multi-key homomorphic encryption, Attribute-based encryption, Access control.**

I. Introduction

NOWDAYS wearable devices are equipped with a wide range of sensors to collect various types of data from users, such as data related to their health (e.g., heart rate, oxygen saturation), activities (e.g., steps count, sleep quality), and environment (e.g., location, humidity levels) [1]. Modern healthcare systems can utilise this data to run analytic models, which could then be used to improve services for (i) individuals, e.g., personalised treatments, remote patient monitoring, and early disease diagnosis by detecting anomalies, and (ii) the wider public, e.g., predicting the spread of disease by analysing data from multiple individuals [2].

Data requesters such as organisations (e.g., healthcare providers and research institutions) and individuals (e.g., family members, friends, and data owners themselves), could also benefit from having access to the results of these analytic models. Therefore, health systems should be able to support data processing and sharing in the following three cases: (i) data owners (DO) accessing the results of analytics run on their own data (DO-DO), and (ii) data requesters (DRs) accessing the results of analytics run on data of a single data owner (DRs-DO), or (iii) on data of multiple data owners (DRs-DOs). These cases are shown in Fig. 1. Moreover, due to the resource-constrained nature of wearable devices on the DO side, there is a need to address the above three cases in an efficient manner.

Healthcare provision via wearable devices has led to a large number of applications deploying cloud service providers (CSP) to run these analytic models. This comes with concerns over user privacy. First, although collected through secure channels, service providers typically process data in plaintext. This comes with risks of data leaks to unauthorised parties [3]. Second, users usually have no control over who has access to their data [4]–[6]. Note that unauthorised exposure of personal health data also violates GDPR [7] and HIPPA [8] regulations, which advocate for users' privacy protection and access control. Hence, it is essential to achieve flexible data processing and sharing in a privacy-preserving manner and efficiently support the three use cases while adopting a user-centric approach. Such an approach protects users' data from unauthorised access and gives control over the data in the hands of data owners rather than service providers [9].

There are already attempts in the literature to achieve privacy-preserving processing and sharing of data, mainly based on homomorphic encryption (HE) or attribute-based encryption (ABE) schemes. However, none of the existing solutions can support all three cases: DO-DO, DRs-DO, and DRs-DOs. Most existing solutions for secure flexible data processing are mainly based on single-key HE schemes [10]–[16], which come with drawbacks. First, user data is encrypted with the same public key that belongs either to one

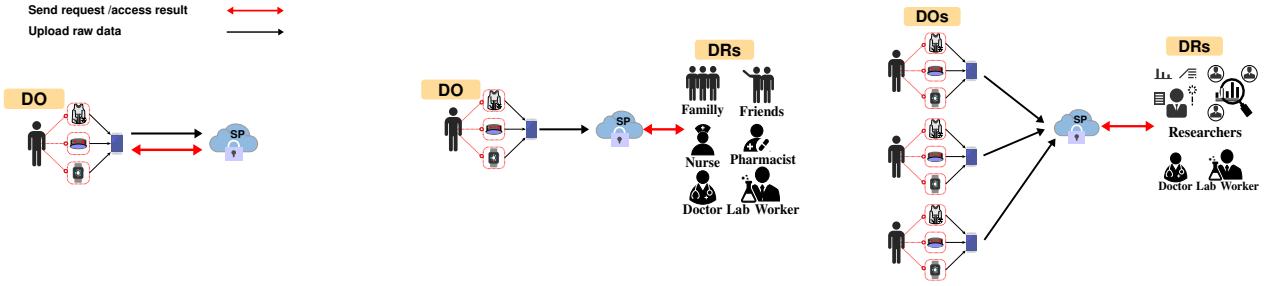
K. Jastaniah is supported by The University of Jeddah funded by the Saudi Government Scholarship. M. A. Mustafa is supported by the EPSRC through the projects EnnCore EP/T026995/1 and SCorCH EP/V000497/1.

K. Jastaniah, N. Zhang, and M.A. Mustafa are with the Department of Computer Science, The University of Manchester, Manchester, UK, e-mail: {khlood.jastaniah, ning.zhang, mustafa.mustafa}@manchester.ac.uk.

K. Jastaniah is also with the College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia.

M.A. Mustafa is also with COSIC, KU Leuven, Leuven, Belgium.

Manuscript received xxx xx, 2021; revised xxx xx, 2021.



(a) A data owner requests aggregation of own data (DO-DO).

(b) Data requesters request aggregation data of a single data owner (DRs-DO).

(c) Data requesters request aggregation data of multiple data owners (DRs-DOs).

Fig. 1: Data aggregation requests use cases.

of the data requesters or to a third-party cloud provider. If the corresponding private key is leaked, this can lead to serious privacy issues [17], [18]. Second, these solutions restrict DOs from accessing and retrieving their data as they cannot access the corresponding private key [19], hence cannot fulfil the DO-DO case. In addition, all HE-based solutions by default do not generally support flexible data sharing with multiple data requesters [20], [21]. A trivial solution to overcome these drawbacks would be for each data requester and data owner to encrypt their data with the homomorphic public key of the requester. However, this is an extremely inefficient solution as it comes with additional costs for the data owner.

To address these drawbacks, a few multi-key HE solutions [22]–[24] have been proposed. Some are based on fully HE (FHE) schemes, which are considered unsuitable for resource-constrained devices, while others are based on partial HE (PHE) schemes. Although they address the data owners access case (DO-DO), none support fine-grain data sharing capabilities with multiple requesters [10], [18], thereby failing to accommodate the remaining two cases – DRs-DO and DRs-DOs. Solutions based on ABE [5], [6], [25]–[27], on the other hand, provide secure data sharing with multiple requesters. However, ABE is intended for data sharing, so they do not support secure data processing. Hence, none of the approaches on its own can support both secure and flexible data processing and sharing.

The solutions proposed in [20], [28], and [29] combine PHE and ABE schemes to address the DRs-DOs case. However, these solutions use a single-key HE scheme, bearing all the drawbacks of such schemes mentioned earlier. Moreover, despite supporting flexible data sharing through ABE, they fail to support personalised processing and access control settings by data owners. They leave this control to third-party cloud servers. In summary, achieving data aggregation to support the three cases (DO-DO, DRs-DO, and DRs-DO) is challenging and not straightforward. Specifically, the scheme should support aggregation of data coming from the same user (encrypted with the same key), as well as aggregation of data coming from different users (encrypted with different keys) without adding burden on the data owner side. Therefore, supporting privacy-preserving data processing along with fine-grain sharing that encompass the three cases (DO-DO, DRs-DO, DRs-DOs) shown in Fig. 1 in an efficient way (i.e., suitable for resource-constrained devices) is still an open issue.

To fill this research gap, we propose a secure, flexible, privacy-preserving and user-centric single and multiple data owners data aggregation and sharing scheme, named SAMA. Compared to the existing solutions, the main novelty (difference) lies in the ability of SAMA to support multi-key PHE. To the best of our knowledge, SAMA is the first scheme to combine multi-key PHE with ABE that supports flexible privacy-preserving data aggregation along with fine-grain sharing with a focus on user-centric access control, and yet it is suitable for resource-constrained devices. To this end, the novel contributions of this work are two-fold:

- We propose a novel scheme called SAMA that combines multi-key PHE and CP-ABE to offer flexible data aggregation with fine-grain sharing capabilities. SAMA addresses data aggregation and fine-grain sharing needs of modern healthcare applications in an efficient manner suitable for resource-constrained devices where data owners upload their data only once and yet supports all three use cases: DO-DO, DRs-DO, and DRs-DOs. It enables data owners to access their raw data and aggregation results and to set access control to their outsourced data with two different access policies. In essence, SAMA accommodates all three cases simultaneously in a single scheme, which has not been addressed previously. It achieves this by performing masking/demasking of (encrypted) data at different stages on the server side based on the nature of the request for data. This allows it to remain efficient on the user side (each data item is encrypted only once) while supporting all three cases and ensuring that data owners have control over who has access to their data.

- We investigate SAMA both theoretically, in terms of security and complexity, and experimentally, in terms of computational and communication costs through simulations. Our results show that SAMA is more efficient than the related works in terms of computational and communication costs.

The rest of the paper is organised as follows. Section II discusses related work. Section III introduces design preliminaries and the main building blocks used. Section IV details the design of SAMA. Sections V and VI detail SAMA's security analysis and performance evaluation, respectively. Section VII concludes the paper.

II. Related Work

There are efforts to protect users' privacy while their data is being processed using single- or multi-key HE schemes. Most solutions with single-key PHE schemes [12]–[14], [16] limit data owners' access to data, and might cause privacy issues. They have also considered secure data processing provided only by a single user or multiple users. Other schemes [22]–[24] are based on FHE schemes, which are not suitable for resource-constrained devices [10]. A few multi-key PHE schemes [10], [18] may be suitable for resource-constrained devices. However, they lack flexible data sharing capabilities. Therefore, multi-key PHE schemes alone are not suitable for addressing privacy-preserving data processing and sharing.

To support secure access control, there are proposals [5], [6], [25]–[27], [35] adopting ABE schemes [36]. These proposals allow users to choose who can access their data, hence supporting fine-grain access control and multiple data requesters' access. ABE schemes can be classified into two types: ciphertext-policy ABE (CP-ABE) [37] and key-policy ABE (KP-ABE) [38] schemes. In the CP-ABE scheme, the access structure is embedded with ciphertexts, and users' attributes are embedded with the users' private keys. In contrast, the KP-ABE scheme's access structure is associated with users' private keys, and the ciphertext is associated with attributes. Therefore, with the KP-ABE schemes, users do not have control over who can access the data; they can only control attribute assignments [37]. In summary, ABE schemes on their own do not support computations over encrypted data.

There are some existing proposals that combine secure data processing with access control. Ding et al. [20], [39], and Wang et al. [29] proposed flexible access control schemes over encrypted user's data computation results by combining ABE with single-key PHE techniques. Their schemes are suitable for resource-constrained devices. However, they cannot support data owners' access their data (DO-DO) as data is encrypted with a cloud key; thereby, they do not provide flexible processing for all three use cases despite providing a fine-grain sharing option. Ruj and Nayak [30] and Chen et al. [28] combined Paillier HE with ABE to support privacy-preserving data aggregation and access control in a smart grid. However, the aggregated data needs to be decrypted and then re-encrypted with an access policy by a trusted authority in Ruj [30]. Chen et al. [28] extended the scheme by incorporating the BLS signature. Mustafa et al. [21] designed a multi-recipient system called DEP2SA, which combines HE and selective data aggregation in a smart grid. Zhang et al. [32] proposed a privacy-preserving data aggregation scheme with fine-grain access control for the smart grid, which combined HE with proxy re-encryption. Liu et al. [34] designed a scheme to support a secure computation of aggregate statistics using Multi-Party Computation (MPC) with fine-grain access control of outsourcing IoT data using CP-ABE. However, using the MPC technique requires high user active interactions for processing and more communication overhead as the number of the generated shares depends on the number of fog nodes. These schemes support only the DRs-DOs case and fail to support the other two use cases. Moreover, they do not support a user-centric access policy.

Bhowmik and Banerjee [31] developed an efficient privacy-preserving aggregation method in which data is aggregated at edge servers en route to a cloud server. Then, the aggregation results are accessed by authorised medical professionals. Tang et al. [33] proposed privacy-preserving fog-assisted health data sharing that supports a flexible user-centric approach using ABE. However, this scheme requires heavy processing at the user side, which might not be suitable for resource-constrained devices. In addition, both of these schemes support only the DRs-DO use case.

Pang and Wang [10] and Zhang et al. [18] proposed privacy-preserving operations on outsourced data from multiple parties in multi-key environments. The proposals support sharing of the processed data only with a data requester, thereby supporting only the DO-DO use case. Moreover, they do not support fine-grain data sharing with multiple DRs.

In summary, most existing solutions that address privacy-preserving data processing do not offer fine-grain data sharing. A few solutions combining both cannot efficiently accommodate all three use cases, as shown in Table I. Therefore, we aim to address this research gap by designing a scheme that efficiently supports flexible, secure data aggregation with fine-grain sharing to accommodate all three use cases under user-centric access control that offers two access control policies (AP_s) and (AP_m) to DOs.

III. Preliminaries

A. System Model

The system model used by SAMA consists of the following entities (see Fig. 2). *Data owners (DO)* are individuals who possess *wearables* and are willing to process their data

TABLE I: Comparison of SAMA with related work in use-cases support.

	[30]	[21]	[31]	[32]	[33]	[10]	[18]	[28]	[29]	[34]	[20]	SAMA
DO-DO support	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	✗	✓
DRs-DO support	✗	✗	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓
DRs-DOs support	✓	✓	✗	✓	✗	✗	✗	✓	✓	✓	✓	✓

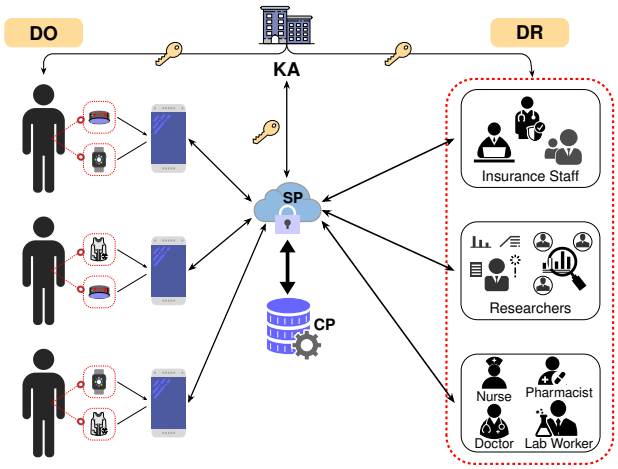


Fig. 2: System model of the SAMA scheme.

TABLE II: Acronyms

Acronym	Meaning
DO	Data Owner
DR	Data Requester
SP	Service Provider
CP	Computation party
KA	Key Authority
HE	Homomorphic Encryption
FHE	Fully Homomorphic Encryption
PHE	Partial Homomorphic Encryption
PE	Paillier Encryption
VP-HE	Variant Paillier Homomorphic Encryption
ABE	Attribute Based Encryption
CP-ABE	Ciphertext-Policy Attribute-Based Encryption
DO-DO	Data owner access own agg. result
DRs-DO	Data requesters access single data owner agg. result
DRs-DOs	Data requesters access multiple data owners agg. result
SAMA	Single And Multiple (DO(s)) Data Aggregation

for their own personal benefits or the collective benefit of society and share their data and processing results with various *data requesters*. Data owners' wearable data is usually collected and shared via their *smartphone (gateway)*. *Data requesters* (DRs) are data consumers who wish to utilise data owners' wearable data in order to provide personalised services to data owners or society. Example DRs could be individuals (e.g., data owners themselves, family members, friends) and organisations (e.g., hospitals, research centres). *Service provider* (SP) provides data storage, manages data access requests, and processes data owners' data, while *Computational party* (CP) cooperates with SP in data computations and access control. A *Key Authority (KA)* plays the role of a key management organisation.

The list of acronyms used throughout the paper is given in Table II.

B. Threat Model

The threat model of the proposed SAMA scheme is as follows: DOs, DRs, SP, and CP are semi-honest (honest-but-curious) entities. They follow the protocol as per the specifications, yet they are curious about the sensitive information of data owners or their data aggregation. The KA is considered a trustworthy entity. It performs all its duties honestly and never colludes with the other entities. The external entities are considered to be untrustworthy, hence malicious. They may utilize different network eavesdropping attacks, modify data in transit, or try to gain unauthorised access to disrupt the system.

Herein, we define the capabilities of adversary $\mathcal{A}$ as follows: $\mathcal{A}$ may compromise the SP and try to guess the raw data from the encrypted data sent by DOs/CP, or to be sent for DRs. $\mathcal{A}$ may also comprise one or more DOs, targeting to guess the processing result in the DRs-DOs case. Similarly, $\mathcal{A}$ may compromise one or more DRs, targeting to obtain the raw data of DOs in the DRs-DOs case. However, adversary $\mathcal{A}$ is restricted from compromising the CP, and the challenged DO and DRs in the DO-DO and DRs-DO cases. If $\mathcal{A}$ compromises the DO, it can gain access to plaintext. Similarly, if $\mathcal{A}$ compromises the DRs, the decrypted processing result of the DO can be obtained. If $\mathcal{A}$ compromises the CP, it can access the strong secret key and thereby access all the raw data from DOs. This threat model is representative and common among adversaries used in other state-of-the-art schemes [10] and [20].

C. Assumptions

We consider the following assumptions in our design.

- The communication channels among all entities are encrypted and authenticated.
- The SP and CP do not collide with each other and any other entities or external adversaries as they have a legal responsibility to prevent leakage of the DOs' sensitive data.
- All entities' identities are verified by KA before obtaining their cryptographic public/private keys.

D. Design Requirements

The proposed system should satisfy the following functional, security and privacy, and performance requirements.

1) Functional Requirements:

- Flexible data processing requests: SAMA should support all of the following three use cases: (i) data owners accessing the aggregation results of their own data (DO-DO), (ii) data requesters accessing the aggregation results of a single data owner (DRs-DO), and (iii) of multiple data owners (DRs-DOs).
- Fine-grain access control: SAMA should support a flexible access policy for data owners and facilitate granting different access rights to a set of data requesters based on the access policy set by data owners.
- User-centric: each data owner should control who is authorised to access the raw data collected from their wearables as well as the aggregated data that contains their raw data in (DRs-DO) and (DRs-DOs) use cases.

2) Security and Privacy Requirements:

- Data confidentiality: data owner's raw and aggregated data should be protected from unauthorised disclosure in transit, at rest, and while processing.
- Authorisation: Only authorised data owners should be able to access data owners aggregated data.

3) Performance Requirements:

- Efficiency: SAMA should be viable for devices with limited computational capabilities.

E. Building Blocks

This section reviews briefly the Paillier cryptosystem [40], the Variant-Paillier in Multi-key cryptosystem [10], and CP-ABE [37], which are used in the design of SAMA. The notations used in the paper are given in Table III.

1) Paillier Cryptosystem: It is a practical and semantically secure additive homomorphic encryption scheme [40].

Paillier in Single-Key Environment consists of three algorithms: key generation algorithm (KGen_{PE}), the encryption algorithm(Enc_{PE}), and decryption algorithm(Dec_{PE}).

- $\mathsf{KGen}_{PE}(k) \rightarrow ppk, psk$: Given a security parameter k, select two large prime numbers p and q. Compute $\mathsf{n} = \mathsf{p}\cdot\mathsf{q}$, and $\lambda = lcm(\mathsf{p}-1, \mathsf{q}-1)$. Define $L(x) = (x-1)/\mathsf{n}$. Select a generator $\mathsf{g} \in \mathbb{Z}^*_{\mathsf{n}^2}$. Compute $\mu = (L(\mathsf{g}^\lambda \mod \mathsf{n}^2))^{-1} \mod \mathsf{n}$. The public key is $ppk = (\mathsf{n},\mathsf{g})$ and the private key is $psk = (\lambda, \mu)$.
- $\mathsf{Enc}_{PE}(ppk, \mathsf{m}) \rightarrow \mathsf{c}$: Given a message $\mathsf{m} \in \mathbb{Z}$ and a public key $ppk = (\mathsf{n},\mathsf{g})$, choose a random number $\mathsf{r} \in \mathbb{Z}^*_{\mathsf{n}}$, and compute $\mathsf{c} = \mathsf{Enc}_{PE}(ppk, \mathsf{m}) = \mathsf{g}^{\mathsf{m}} \cdot \mathsf{r}^{\mathsf{n}} \mod \mathsf{n}^2$.
- $\mathsf{Dec}_{PE}(psk, \mathsf{c}) \rightarrow \mathsf{m}$: Given a ciphertext c and a private key $psk = (\lambda, \mu)$, recover the message $\mathsf{m} = \mathsf{Dec}_{PE}(psk, \mathsf{c}) = L(\mathsf{c}^\lambda \mod \mathsf{n}^2) \cdot \mu \mod \mathsf{n}$.

Variant-Paillier in Multi-Key Environment [10] is a variation of the Paillier cryptosystem [40], which makes it compatible to work in multiple users (DOs) environment by generating a different public-private key pair for each data owner with two trapdoor decryption algorithms. The scheme comprises: key generation (KGen_{VP}), encryption (Enc_{VP}), decryption with a weak secret key (Dec_{wsk}), and decryption with a strong secret key (Dec_{ssk}).

- $\mathsf{KGen}_{VP}(k) \rightarrow vpk, wsk, ssk$: Given a security parameter k, choose $k+1$ small odd prime factors $u, v_1, \ldots, v_i, \ldots, v_k$ and choose two large prime factors v_p and v_q in which p and q are large primes with the same bit length. Compute p and q as $p = 2uv_1v_2\cdots v_i\cdots v_kv_p + 1$ and $q = 2uv_1v_2\cdots v_i\cdots v_kv_q + 1$. Calculate $n = p\cdot q$ and $\lambda = lcm(p-1, q-1)$. Choose t as a number or a product of multiple numbers from the set $(v_1, v_2, \ldots, v_i, \ldots, v_k)$, and $t|\lambda$ naturally exists. Choose a random integer $g \in \mathbb{Z}^*_{n^2}$ that satisfies $g^{utn} = 1 \mod n^2$, and $\gcd(L(g^\lambda \mod n^2), n) = 1$. Define $L(x) = (x-1)/n$. Compute $h = g^{n\times\lambda/t} \mod n^2$. The public key is $vpk = (n, g, h)$, the weak secret key is $wsk = t$ and the strong secret key is $ssk = \lambda$
- $\mathsf{Enc}_{VP}(vpk, m) \rightarrow c$: Given a message $m \in \mathbb{Z}_n$ and a public key $vpk = (n, g, h)$, choose a random number $r \in \mathbb{Z}_n$, and compute the ciphertext c as $c = \mathsf{Enc}_{VP}(vpk, m) = g^m h^r \mod n^2$.
- $W\mathsf{Dec}_{VP}(wsk, c) \rightarrow m$: The decryption algorithm with a weak secret key decrypts only the ciphertext encrypted with the associated public key. Given wsk and c, the ciphertext can be decrypted as $m = W\mathsf{Dec}_{VP}(wsk, c) = \frac{L(c^t \mod n^2)}{L(g^t \mod n^2)} \mod n$.
- $S\mathsf{Dec}_{VP}(ssk, c) \rightarrow m$: The decryption algorithm with a strong key decrypts the ciphertexts encrypted with any public key of the scheme. Given ssk and c, the ciphertext can be decrypted as $m = S\mathsf{Dec}_{VP}(ssk, c) = L(c^\lambda \mod n^2)\cdot\mu \mod n$.

$$\frac{L(c^\lambda \mod n^2)}{L(g^\lambda \mod n^2)} \mod n = \frac{L(g^{\lambda m} \mod n^2)}{L(g^\lambda \mod n^2)} \mod n$$

2) Ciphertext-Policy Attribute Based Encryption: The CP-ABE is a type of public-key encryption in which the ciphertext is associated with an access policy and data owner's private key is dependent upon attributes to support fine-grain access control [10]. It consists of four main algorithms: a setup algorithm ($Setup$), encryption algorithm (Enc_{ABE}), key generation algorithm (KGen_{ABE}), and decryption algorithm (Dec_{ABE}).

- $Setup(k, U) \rightarrow \mathsf{pk}, mk$: Given a security parameter k and a universe of attributes U, the setup algorithm outputs the public parameters pk and a master key mk.
- $\mathsf{Enc}_{ABE}(\mathsf{pk}, M, A) \rightarrow C$: Given public parameters pk, a message M, and an access structure A over the

TABLE III: Notations

Symbol	Meaning
DO_i	ith DO, $i = \{1, \ldots, N\}$
m_i	raw data provided by DO_i
r_i	random number generated by SP for each DO_i
N_{DO}, N_{DR}	number of DOs, DRs
N_{req}	number of data points requested for aggregation
N_m	number of messages received by DO_i
N	number of DOs in DRs-DOs case
ssk	strong secret key in VP-HE
vpk_i, wsk_i	VP-HE key pair (public key, weak secret key) of DO_i
Enc_{VP}, Dec_{VP}	encryption, decryption using VP-HE
$[m_i]$	m_i encrypted by the vpk_i of DO_i
$[r_i]$	random number encrypted by the vpk_i of DO_i
Enc_{PE}, Dec_{PE}	encryption, decryption using PE
ppk_j, psk_j	PE public, private key pair used by DR
pk, MK, sk	public parameters, master key, secret key in CP-ABE
Enc_{ABE}, Dec_{ABE}	encryption, decryption using CP-ABE
AP_S/AP_M	single- and multiple-$DO(s)$ data access policy
$BiPair$	cost of bilinear pairing in ABE
$\lvert\gamma\rvert + 1$	number of attributes in the access policy tree
ϑ	number of attributes needed to satisfy the access policy

universe of attributes, the encryption algorithm outputs a ciphertext C which implicitly contains A.

- $\mathsf{KGen}_{ABE}(mk, s) \rightarrow \mathsf{sk}$: Given a master key mk and a set of attributes s which describe the key, the key generation algorithm outputs a private key sk.
- $\mathsf{Dec}_{ABE}(\mathsf{pk}, C, \mathsf{sk}) \rightarrow M$: Given public parameters pk, a ciphertext C, which includes an access policy A, and a private key sk, using a decryption algorithm, a data owner can decrypt the ciphertext and get a message M only if the attributes associated with the private key satisfy A.

IV. The SAMA Scheme

A. Overview of the SAMA Scheme

SAMA combines the VP-HE and CP-ABE schemes and consists of three phases: (i) DO access policy setting, (ii) data uploading, and (iii) data access request and processing. The overview of these phases is shown Fig. 3.

During the first phase, to achieve a user-centric fine-grain access policy functionality, data owners define two types of policies, single (AP_S) and multiple (AP_M) access policies, and send them to SP. This allows SP to process and share DOs' data with multiple DRs according to DOs' preferences. In the next phase, DOs encrypt data only once with their VP-HE public key and send the resulting ciphertext to SP. During the last phase, SP receives requests to grant access to the (aggregated) data of DOs. Both SP and CP process these requests, and the results are shared with the requester. There can be three types of requests: from DOs for accessing aggregation of their own data (DO-DO) [13], [14] or from the DRs requesting (aggregated) data of either a single DO (DRs-DO) [31], [33] or multiple DOs (DRs-DOs) [20], [21], [29].

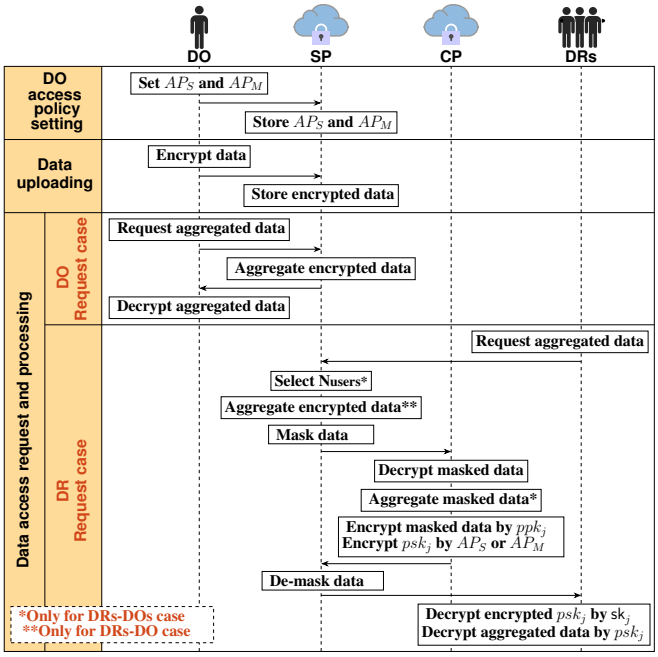


Fig. 3: An overview of the SAMA scheme.

Upon receiving a request from a DO, SP aggregates the DO's encrypted data and returns the result. The DO uses its own VP-HE weak secret key to obtain their aggregated result. If the request comes from a DR for aggregation of a single DO's data, SP aggregates the DO's encrypted data, masks it, and sends the result to CP. CP performs strong decryption to obtain the masked data, encrypts this result (masked aggregated data) with a Paillier public key and encrypts the corresponding private key using CP-ABE with AP_S, and sends both ciphertexts to SP. If the request is for data of multiple DOs, the process is slightly different. SP gets the encrypted data of DOs, masks them, and sends the masked encrypted data to CP. CP performs strong decryption on the received ciphertexts, aggregates the results (masked data), encrypts the result with a Paillier public key, and encrypts the corresponding private key using CP-ABE with AP_M. In both cases, CP sends both ciphertexts to SP. SP then performs de-masking on the received ciphertext and sends the encrypted result (aggregated data) and CP-ABE ciphertext to DR. Finally, DRs who satisfy the access policy decrypt CP-ABE ciphertext and obtain the Paillier private key to access DOs aggregate data.

B. System Initialisation

1) System Parameters Setup: In this phase, system parameters of the encryption schemes (which determine the key lengths) are set.

- VP-HE setup: The KA sets a security parameter k and chooses two large prime numbers p and q such that $L(p) = L(q) = k$. L is the bit length of the input data.
- Paillier setup: Given the security parameter k, the KA then chooses two large prime numbers p and q. Then, the key generation algorithm is initiated as in Section. III-E1
- ABE setup: Given the security parameter k, the KA generates U attributes, which are used to generate pk and mk using the $Setup$ algorithm described in Section III-E2.

2) System Key Generation and Distribution: This phase is divided into three steps outlined below.

- *V*P-HE Key Generation: The KA generates a unique ssk and distinct variant Paillier homomorphic public/private key pair (vpk_i, wsk_i) for every DO, DO_i, $i = 1, \ldots, N_{DO}$, using the $KGen_{VP}$ algorithm in Section. III-E1.
- Paillier Key Generation: The KA generates a distinct Paillier homomorphic public/private key pair (ppk_j, psk_j), for each request that comes from the same or any DR, using the $KGen_{PE}$ algorithm described in Section. III-E1.
- ABE Key Generation: The KA generates a distinct private key sk_j for every DR_j, using $KGen_{ABE}$ (Section. III-E2). DR_j obtains sk_j, which embeds her/his attributes/roles.

C. SAMA in Detail

SAMA consists of three phases: DO access policy setting, data uploading, and data access request and processing.

1) DO Access Policy Setting: It allows data owners to set their access policy for data aggregation and sharing requirements and share it with SP. It has three steps: a) define the access policy, b) activate notifications, and c) update the access policy.

a) Define access policy: Each data owner defines two types of access policy: AP_S – single-data owner and AP_M – multiple-data owners data aggregation and sharing access policy. AP_S allows data owners to control who can access the aggregated results of their own data. Only authorised data requesters with specific attributes satisfying the access policy can have access to the final aggregated result. AP_M allows data owners to set whether they agree their data to be aggregated with other data owners' data and the result to be shared. Each data owner defines their sharing preferences and gives consent to allow the use of their data in aggregation along with other data owners' data. AP_M does not authorise SP to share any specific individual raw data with anyone. It only allows SP to use the encrypted data of data owners whose sharing preferences match with the attributes of data requesters who requested data access.

b) Activate notification: data owners can select to receive regular notifications: a summary of all requests for their data received by SP. They can check who has requested access to their data and whose requests were granted/rejected. Regular notifications can be switched on/off by data owners and be received as daily/weekly/monthly summaries.

c) Update access policy: SP provides data owners with the ability to update their access policy periodically or on demand. Data owners can also update their pre-defined policies (AP_S or AP_M).

2) Data Uploading: During this phase, DOs upload their data to SP regularly. They encrypt their wearable data m_i with their variant-Paillier public key, vpk_i, to obtain $C_{vpk_i} = \mathsf{Enc}_{VP}(vpk_i, m_i)$ and send the encrypted data to SP. This phase is the same for single and multi data owners data sharing.

3) Data Access Request and Processing: There can be three different types of data access requests for data owners' aggregated data: a) Data owner requests access to their own (aggregated) data (DO-DO), and DRs request access to aggregated data of b) a single DO (DRs-DO) and c) multiple DOs (DRs-DOs). DO requests are directly handled by SP, while DR requests are handled by both SP and CP.

a) DO-DO use case: A DO requests SP to aggregate their own encrypted wearable data and provide the result. Upon receiving the request to aggregate N_{req} data points, SP aggregates the data owner's data (i.e., it performs additive homomorphic operations by multiplying the encrypted data owner's data) to get $[\sum_{i=1}^{N_{req}} m_i]_{vpk_i} = \prod_{i=1}^{N_{req}} C_{vpk_i}$, where $[m]$ denotes encrypted data. The result then is sent to the DO who can decrypt $[\sum_{i=1}^{N_{req}} m_i]_{vpk_i}$ with his weak secret key to get the aggregated data, $\sum_{i=1}^{N_{req}} m_i \ mod \ n = W\mathsf{Dec}_{VP}(wsk_i, [\sum_{i=1}^{N_{req}} m_i]_{vpk_i})$. For simplicity and improved readability, in the rest of the paper, we will omit $mod\ n$ in the aggregation results.

b) DRs-DO use case: A DR requests access to the aggregated data of a single DO, e.g., a doctor requires access to the aggregated data of a patient. The aggregated data can be accessed only by DRs whose attributes satisfy the fine-grain access policy AP_S set by the DO

(i) Handling DR request: After DR has issued a request to access the aggregated data, SP performs the same additive homomorphic operations, as in Step a) explained earlier. The result is a ciphertext of the aggregated data: $[\sum_{i=1}^{N_{req}} m_i]_{vpk_i}$.

(ii) Masking: SP masks the aggregated data. Specifically, it generates a random number and encrypts it with the data owner's VP-HE public key, vpk_i, $[r_i]_{vpk_i} = \mathsf{Enc}_{vp}(vpk_i, r_i)$. The ciphertext is then multiplied with the ciphertext of the aggregated data to get a ciphertext of the masked aggregated data, $[\sum_{i=1}^{N_{req}} m_i + r_i]_{vpk_i} = [\sum_{i=1}^{N_{req}} m_i]_{vpk_i} * [r_i]_{vpk_i}$. The result is sent to CP along with the AP_S set by the data owner.

Algorithm 1 SAMA: DRs-DO use case

Input: Two ciphertexts $[m_1]_{vpk_i}$ and $[m_2]_{vpk_i}$
Output: The ciphertexts $[m_1 + m_2]_{ppk_j}$
Steps by SP:
$[m_1 + m_2]_{vpk_i} \leftarrow [m_1]_{vpk_i} \times [m_2]_{vpk_i}$
$r_i \xleftarrow{R} \mathbb{Z}_n$
$[r_i]_{vpk_i} \leftarrow \mathsf{Enc}_{vp}(vpk_i, r_i)$
$[m_1 + m_2 + r_i]_{vpk_i} = [m_1 + m_2]_{vpk_i} \times [r_i]_{vpk_i}$
Send $[m_1 + m_2 + r_i]_{vpk_i}$ and AP_S to CP
Steps by CP:
$(m_1 + m_2 + r_i) \leftarrow S\mathsf{Dec}_{VP}(ssk, [m_1 + m_2 + r_i]_{vpk_i})$
$[m_1 + m_2 + r_i]_{ppk_j} \leftarrow \mathsf{Enc}_{PE}(ppk_j, (m_1 + m_2 + r_i))$
$[psk_j]_{AP_S} \leftarrow \mathsf{Enc}_{ABE}(\mathsf{pk}, psk_j, AP_S)$
Send $[m_1 + m_2 + r_i]_{ppk_j}$ and $[psk_j]_{AP_S}$ to SP
Steps by SP:
$[r_i]_{ppk_j} = \mathsf{Enc}_{PE}(ppk_j, r_i)$
$[-r_i]_{ppk_j} \leftarrow ([r_i]_{ppk_j})^{n-1}$
$[m_1 + m_2]_{ppk_j} = [m_1 + m_2 + r_i]_{ppk_j} \times [-r_i]_{ppk_j}$
Send $[m_1 + m_2]_{ppk_i}$ and $[psk_j]_{AP_S}$ to DR

(iii) Preparing the processing result: CP decrypts the result using its strong decryption key to get the masked aggregate data, $\sum_{i=1}^{N_{req}} m_i + r_i = S\mathsf{Dec}_{VP}(ssk, [\sum_{i=1}^{N_{req}} m_i + r_i]_{vpk_i})$. A new Paillier key pair (ppk_j, psk_j) is generated by KA and sent to CP. ppk_j is used to encrypt the masked aggregated data to get $[\sum_{i=1}^{N_{req}} m_i + r_i]_{ppk_j} = \mathsf{Enc}_{PE}(ppk_j, \sum_{i=1}^{N_{req}} m_i + r_i)$, while the psk_j is encrypted by the DO defined access policy (AP_S) to get $[psk_j]_{AP_S} = \mathsf{Enc}_{ABE}(\mathsf{pk}, psk_j, AP_S)$. Both ciphertexts are sent to SP.

(iv) De-masking: SP initiates a de-masking process. It encrypts the random number r_i (used in the masking process) with ppk_j to obtain $[r_i]_{ppk_j} = \mathsf{Enc}_{PE}(ppk_j, r_i)$. Then, it calculates the additive inverse of $[r_i]_{ppk_j}$, generating $[-r_i]_{ppk_j} = [r_i]_{ppk_j}^{n-1}$. Finally, it de-masks the aggregated data: $[\sum_{i=1}^{N_{req}} m_i]_{ppk_j} = [\sum_{i=1}^{N_{req}} m_i + r_i]_{ppk_j} * [-r_i]_{ppk_j}$. Finally, SP sends $[\sum_{i=1}^{N_{req}} m_i]_{ppk_j}$ and $[psk_j]_{AP_S}$ to DR.

(v) DR access the processing result: DR can access the processing result only if the DR's key attributes satisfy the data owner's AP_S. Hence, DR can decrypt and obtain psk_j by using its ABE secret key $psk_j = \mathsf{Dec}_{ABE}(\mathsf{pk}, [psk_j]_{AP_S}, \mathsf{sk})$. Finally, it uses psk_j to obtain the initially requested aggregated data of the data owner : $\sum_{i=1}^{N_{req}} m_i = \mathsf{Dec}_{PE}(psk_j, [\sum_{i=1}^{N_{req}} m_i]_{ppk_j})$.

c) DRs-DOs use case: A DR requests access to aggregated data of multiple DOs. The aggregated data can be accessed only by DRs whose attributes satisfy the AP_M (see

(i) Handling DR request: Upon receiving a data access request, SP initiates the process by comparing data owner's AP_M with DR attributes. It then selectsdata owners whose AP_M matches with the DR request. Let us assume SP selects N data owners.

(ii) Masking: SP generates a random number for every DO's data and encrypts them with the corresponding DO's variant Paillier public key, $[r_i]_{vpk_i} = \mathsf{Enc}_{vp}(vpk_i, r_i)$. Next, each encrypted random number is multiplied with the respective data owner's encrypted data to obtain $[m_i + r_i]_{vpk_i} = [m_i]_{vpk_i} * [r_i]_{vpk_i}$. Finally, the N masked ciphertexts are sent to CP with the AP_M set by the data owner for further processing.

Algorithm 2 SAMA: DRs-DOs use case.

Input: Two ciphertexts $[m_1]_{vpk_1}$ and $[m_2]_{vpk_2}$
Output: The ciphertexts $[m_1 + m_2]_{ppk_j}$
Steps by SP:
$r_1, r_2 \xleftarrow{R} \mathbb{Z}_n$
$[r_1]_{vpk_1} \leftarrow \mathsf{Enc}_{vp}(vpk_1, r_1)$
$[r_2]_{vpk_2} \leftarrow \mathsf{Enc}_{vp}(vpk_2, r_2)$
$[m_1 + r_1]_{vpk_1} = [m_1]_{vpk_1} \times [r_1]_{vpk_1}$
$[m_2 + r_2]_{vpk_2} = [m_2]_{vpk_2} \times [r_2]_{vpk_2}$
Send $[m_1 + r_1]_{vpk_1}$, $[m_2 + r_2]_{vpk_2}$, and AP_M to CP
Steps by CP:
$(m_1 + r_1) \leftarrow S\mathsf{Dec}_{VP}(ssk, [m_1 + r_1]_{vpk_1})$
$(m_2 + r_2) \leftarrow S\mathsf{Dec}_{VP}(ssk, [m_2 + r_2]_{vpk_2})$
$(m_1 + r_1 + m_2 + r_2) = (m_1 + r_1) + (m_2 + r_2)$
$[m_1 + r_1 + m_2 + r_2]_{ppk_j} \leftarrow \mathsf{Enc}_{PE}(ppk_j, (m_1 + r_1 + m_2 + r_2))$
$[psk_j]_{AP_M} \leftarrow \mathsf{Enc}_{ABE}(\mathsf{pk}, psk_j, AP_M)$
Send $[m_1 + r_1 + m_2 + r_2]_{ppk_j}$ and $[psk_j]_{AP_M}$ to SP
Steps by SP:
$(r_1 + r_2) = r_1 + r_2$
$[r_1 + r_2]_{ppk_j} \leftarrow \mathsf{Enc}_{vp}(ppk_j, (r_1 + r_2))$
$[-(r_1 + r_2)]_{ppk_j} \leftarrow ([r_1 + r_2]_{ppk_j})^{n-1}$
$[m_1 + m_2]_{ppk_j} = [m_1 + m_2 + r_1 + r_2]_{ppk_j} \times [-(r_1 + r_2)]_{ppk_j}$
Send $[m_1 + m_2]_{ppk_i}$ and $[psk_j]_{AP_M}$ to DR

(iii) Preparing the processing result: First, CP decrypts all the received masked ciphertexts with the variant Paillier strong secret key to obtain the individual DOs' masked data: $m_i + r_i = \mathsf{Dec}_{vp}(ssk, [m_i + r_i]_{vpk_i})$. Then, it performs an addition operation to get the masked aggregation: $\sum_{i=1}^{N} m_i + \sum_{i=1}^{N} r_i = \sum_{i=1}^{N}(m_i + r_i)$. Second, KA generates a new Paillier public-private key (ppk_j, psk_j) for every authorised DR request received. Third, CP encrypts the masked result using the Paillier public key, $[\sum_{i=1}^{N} m_i + \sum_{i=1}^{N} r_i]_{ppk_j} = \mathsf{Enc}_{PE}(ppk_j, (\sum_{i=1}^{N} m_i + \sum_{i=1}^{N} r_i))$, while the corresponding private key psk_j is encrypted with the common AP_M: $[psk_j]_{AP_M} = \mathsf{Enc}_{ABE}(pk, psk_j, AP_M)$. CP sends both ciphertexts $[\sum_{i=1}^{N} m_i + \sum_{i=1}^{N} r_i]_{ppk_j}$, $[psk_j]_{AP_M}$ to SP.

(iv) De-masking: SP aggregates all the random numbers (used for masking) and encrypts the result, $[\sum_{i=1}^{N} r_i]_{ppk_j} = \mathsf{Enc}_{PE}(ppk_j, \sum_{i=1}^{N} r_i)$, computes additive inverse of $[\sum_{i=1}^{N} r_i]_{ppk_j}$, $[-\sum_{i=1}^{N} r_i]_{ppk_j} = [\sum_{i=1}^{N} r_i]_{ppk_j}^{n-1}$, and de-masks the result: $[\sum_{i=1}^{N} m_i]_{ppk_j} = ([\sum_{i=1}^{N} m_i + \sum_{i=1}^{N} r_i]_{ppk_j}) * ([-\sum_{i=1}^{N} r_i]_{ppk_j})$.

(v) DR access the processing result: DR decrypts $[psk_j]_{AP_M}$ using sk if the DR's key satisfies the access policy: $psk_j = \mathsf{Dec}_{ABE}(\mathsf{pk}, [psk_j]_{AP_M}, \mathsf{sk})$. Finally, it uses psk_j to access the data: $\sum_{i=1}^{N} m_i = \mathsf{Dec}_{PE}(psk, [\sum_{i=1}^{N} m_i]_{ppk_j})$.

V. Security Analysis

A. Security of the SAMA Scheme

The security analysis of SAMA is based on the simulation paradigm framework [41] in the presence of semi-honest

(honest-but-curious and non-colluding) adversaries. It is a technique used for comparing adversaries in two worlds: the $REAL$ world, where the adversary has access to the ciphertext and the $IDEAL$ world, where the adversary does not even have access to the ciphertext. Since an $IDEAL$ world adversary does not have access to a ciphertext, they need simulators to generate a ciphertext using random data and keys. Therefore, we construct four simulators (Sim_{DO}, Sim_{SP}, Sim_{CP}, and Sim_{DR}) which represent DO, SP, CP, and DR. They simulate the execution views of the following adversaries Adv_{DO}, Adv_{SP}, Adv_{CP}, and Adv_{DR}. To ensure the security of the SAMA scheme under the simulation paradigm, we need to demonstrate that the execution views of the $REAL$ and $IDEAL$ worlds are indistinguishable. This means that the information learned by an adversary who receives a real ciphertext should be the same as the information learned by an adversary who receives nothing. KA is excluded as it is a trustworthy entity.

$Theorem$ 1. SAMA *can securely retrieve in plaintext the result of the addition operation over encrypted data in the presence of semi-honest adversary models and threat attacks.*

$Proof$: We prove the security of the SAMA scheme by considering the case with two data inputs.

1) Sim_{DO}*:* Sim_{DO} encrypts the provided inputs m_1 and m_2 using VP-HE and returns both ciphertexts to Adv_{DO}. The simulation view of the $IDEAL$ world of Adv_{DO} is computationally indistinguishable from the $REAL$ world view owing to the semantic security of VP-HE.

2) Sim_{SP}*:* Sim_{SP} simulates Adv_{SP} in DRs-DO and DRs-DOs cases. In the former case, Sim_{SP} multiplies the provided ciphertexts and then encrypts a random number r with VP-HE. It then multiplies the encrypted random number with the result of the multiplication of the ciphertexts. Later, this random number is encrypted with the public key of the Paillier scheme, and its ciphertext is raised to $n-1$ and multiplied with the given ciphertext. In the latter case, Sim_{SP} generates two random numbers r_1 and r_2, encrypts them with the VP-HE public key and multiplies the encrypted random numbers with the ciphertexts (encrypted m_1 and m_2), respectively. Later, the same random numbers are encrypted with the Paillier public key, and the results are raised to $n-1$ and multiplied with the given ciphertext. In both cases, the Adv_{SP} receives the output ciphertexts from Sim_{SP}. Therefore, the $REAL$ and $IDEAL$ views of Adv_{SP} are computationally indistinguishable owing to the semantic security of VP-HE and Paillier encryption.

3) Sim_{CP}*:* The execution view of CP in the $REAL$ world is given by both ciphertext of (m_1+r_1) and (m_2+r_2), which are used to obtain m_1+r_1 and m_2+r_2 by executing decryption with the strong secret key on these ciphertexts (r_1 and r_2 are random integers in $\mathbb{Z}_n$). The execution view of CP in the $IDEAL$ world has two ciphertexts randomly selected in the $\mathbb{Z}_{n^2}$. Sim_{CP} simulates Adv_{CP} in both DRs-DO and DRs-DOs cases. In the former case, Sim_{CP} simulates Adv_{CP} as follows. It runs the strong decryption algorithm and obtains $m_1' + m_2' + r'$ and then the decryption result undergoes further encryption by the public key of Paillier scheme to obtain a new ciphertext. In the latter case, Sim_{CP} runs the strong decryption algorithm and obtains $m_1' + r_1'$ and $m_2' + r_2'$. It aggregates the decryption results, and then the aggregated result is further encrypted by the Paillier public key to obtain a ciphertext. In both cases, a randomly generated number is encrypted with CP-ABE. Then, the two ciphertexts (generated by the Paillier and CP-ABE schemes) are provided by Sim_{CP} to Adv_{CP}. These ciphertexts are computationally indistinguishable between the $REAL$ and $IDEAL$ world of Adv_{CP} since the CP is honest and the semantic security of VP-HE and Paillier cryptosystem, and the security of CP-ABE.

4) Sim_{DR}*:* Sim_{DR} randomly selects chosen ciphertexts (besides not having access to challenged data), decrypts, and sends them to Adv_{DR} to gain data information. The view of Adv_{DR} is the decrypted result without any other information, irrespective of how many times the adversary accesses Sim_{DR}. Due to CP-ABE's security and the Paillier scheme's semantic security, both $REAL$ and $IDEAL$ world views are indistinguishable. Since the DO data encryption and DR decryption processes are common for both DRs-DO and DRs-DOs in SAMA, the security proof of Adv_{DO} and Adv_{DR} is common for both cases.

B. Analysis against Security and Privacy Requirements

1) Data Confidentiality: Every data owner encrypts their data using their VP-HE public key vpk_i. SP then performs a homomorphic addition operation over encrypted data and delivers the ciphertext with the encrypted private Paillier key using CP-ABE to DR. Only authorised DRs can obtain the key and hence have access to the DO data. Furthermore, SAMA conceals data owner's raw data by adding random numbers at SP, i.e., masking the data, hence preserving the privacy of the DO(s) data at CP. Moreover, the Paillier cryptosystem is semantically secure, and the CP-ABE is secure under the generic elliptic curve bi-linear group model as discussed in V-A. In addition, the communication channels among all the entities (DO, SP, CP, and DR) are secure (e.g., encrypted using SSL). Therefore, only the authorised entities (i.e., DO/DR) can access the result, and all the unauthorised internal/external entities who might eavesdrop on messages sent and/or collect information can only access ciphertexts.

2) Authorisation: SAMA uses CP-ABE to implement secure access control, where the processing result is encrypted by the data owner's defined access policies and the decryption key is associated with the attributes of the requesters. The user-centric access policy has been applied, which allows data owner to define their access policies to securely and selectively grant DRs access to the processing

result. Thus, the processing result is encrypted using AP_S and AP_M, which are access policies set by data owner to determine their preferences for sharing data processing results. Hence, the private key of DR (sk) is required to decrypt the encrypted processing result using CP-ABE. Only the authorised DR who satisfies the access policy can access the key and decrypt the processing result. Thus, using CP-ABE, SAMA provides user-centric access control, and only authorised data requester can access the processing result.

C. Comparison with Related Work

Table IV provides a comparison of SAMA with the closely related existing scheme that combines both data processing and sharing with respect to our design requirements.

In terms of flexible data processing requests, SAMA is the only scheme that supports all three use cases (DO-DO, DRs-DO, and DRs-DOs), whereas other schemes support only one or two of the use cases. For a complete breakdown of which scheme supports which cases, we refer the reader to Table I.

In SAMA, to achieve fine-grain data sharing, the resource-intensive CP-ABE is outsourced to the cloud, reducing the burden on data owners in a way suitable for resource-constrained devices similar to [20], [32], [33], [42], [28], [29]. Furthermore, like SAMA, the schemes [33], [34] offer a user-centric access policy approach. However, they all provide only one access policy setting option for data owners to control their own data processing and sharing (DRs-DO). Unlike other schemes, SAMA allows data owners to control whether their data is included with other multiple data owners' data (DRs-DOs). All schemes achieve data confidentiality using one of these techniques (homomorphic encryption, MPC, and symmetric encryption). However, the scheme [30] does not achieve end-to-end encryption. For authorisation, all schemes allow only authorised entities to access data.

As discussed earlier, the schemes [20], [30], [31], [28], [29] use PHE schemes in a single-key setting, which requires data to be encrypted with a third party public key, thereby having multiple data owners encrypt their own data with a key that does not belong to them. Hence, these schemes may not be suitable for highly sensitive data applications. Further, using a single-key HE setting does not allow data owners to access their own data directly, as the data is not encrypted with their public keys.

In summary, there are limited efforts to explore the integration of privacy-preserving flexible data processing with fine-grain sharing under user-centric access control [20], [30], [28], [29]. The state-of-the-art research either addresses the issue of privacy preserving data processing and sharing separately or cannot satisfy the diverse demand of modern health systems to accommodate all three use cases efficiently, as shown in Table IV.

VI. Performance Evaluation

In this section, we evaluate SAMA in terms of computational complexity and communication overheads incurred among all entities in the system. We implemented the SAMA scheme and tested its performance through simulation for our evaluation. We also compared the performance of SAMA with the performance of the most relevant solutions [20] and [28]. Note that, for a fair comparison, only these two solutions were chosen as they follow the same approach as SAMA – combining PHE with ABE. The other solutions from Table IV were not included as they utilise other cryptographic primitives such as MPC and proxy re-encryption.

A. Computational Complexity

The computationally expensive operations considered in the SAMA scheme are modular exponentiation and multiplication operations, denoted as $ModExp$ and $ModMul$, respectively. We ignore the fixed numbers of modular additions in our analysis as their computational cost is negligible compared to $ModExp$ and $ModMul$. In our analyses, we also use the following parameters: $BiPair$ – cost of bilinear pairing in ABE; $|\gamma|+1$ – number of attributes in the access policy tree; ϑ – number of attributes needed to satisfy the access policy.

1) Computational Complexity of HE Data Aggregation: In our analysis, we split the computational complexity into four parts: the complexity of each entity.

Computations at DOs: At each reporting time slot, each data owner encrypts their data by their VP-HE public key vpk_i to generate a ciphertext for data processing/analysing. This encryption requires two modular exponentiation operations and one modular multiplication. Hence, the computational complexity at the DO side is $2ModExp + ModMul$.

Computations at SP and CP: This includes operations performed by SP and CP. As these operations are slightly different for the DO-DO, DRs-DO and DRs-DOs cases, we analyse them separately.

For the DO-DO and DRs-DO cases, SP performs additive homomorphic encryption on the received data owner ciphertexts $((N_m - 1)ModMul)$. Then, for the DRs-DO case, it generates a random number r, encrypts it with the DO's VP public key vpk_i $(2ModExp + ModMul)$, multiplies the results of the homomorphic addition with the encrypted random number $(ModMul)$ and sends it to CP. Next, SP re-encrypts the generated random number r by ppk_j $(2ModExp + ModMul)$, calculates the additive inverse of r and then multiplies it with the encrypted processing result to remove the masking from the original data $(ModMul)$. Thus, SP performs total for the DO-DO case: $(N_m - 1)ModMul$, while for the DRs-DO case: $4ModExp + (N_m + 3)ModMul$. In the DRs-DO case, CP performs strong decryption using ssk on the received

TABLE IV: Comparison of SAMA with related work.

Requirement / Feature	[30]	[21]	[31]	[32]	[33]	[10]	[18]	[28]	[29]	[34]	[20]	SAMA
Flexible data processing request	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Fine grain data sharing	✓	✗	✗	✓	✓	✗	✗	✓	✓	✓	✓	✓
User-centric	✗	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✓
Data confidentiality	✓*	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Authorisation	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Multi-key setting	✗	✗	✗	✗	✗	✓	✓	✗	✗	MPC	✗	✓
Trust level on service providers	FT*	ST	ST	ST	ST	ST	ST	ST	ST	ST	ST	ST
Supported operations	ADD	ADD	ADD	ADD	CLASS	MULT, COMP	All	ADD	ADD**	ADD, MULT	All	ADD**

*Agg. results are decrypted by a trusted party.
**It can support more operations but is not included in the paper.
ST - Semi-trusted; FT - Fully-trusted; ADD - Addition; CLASS - Classification; MULT - Multiplication; DIV - Division; COMP - Comparison; All - All operation without classification.

ciphertexts ($ModExp + ModMul$). It then encrypts the aggregated masked result with ppk_j ($2ModExp + ModMul$), and encrypts psk_j with CP-ABE using AP_S ($(|\gamma|+1)Exp$). Hence, CP performs in total: $3ModExp + 2ModMul + (|\gamma| + 1)Exp$. In total, the cost at SP and CP in the DRs-DO case is: $7ModExp + (N_m + 5)ModMul + (|\gamma| + 1)Exp$. Note: CP is not part of the DO-DO case.

For the DRs-DOs case, SP generates a random number for every DO's data, encrypts them using the VP public key of the corresponding DO, vpk_i, ($N_m(2ModExp + ModMul)$), and then multiplies the resulting ciphertexts with the ciphertexts received from DOs ($N_m ModMul$). Later, it aggregates all the generated random numbers, encrypts it using ppk_j ($2ModExp + ModMul$), calculates the additive inverse of the aggregation result, and then multiplies the aggregation result ciphertext with the received ciphertext from CP to remove the masking from the original data ($ModMul$). Thus, the computational cost of SP in the DRs-DOs case is: $(2N_m + 2)ModExp + (2N_m + 2)ModMul$. CP performs strong decryption using ssk for all N received ciphertexts ($N_m(ModExp + ModMul)$), and then aggregates the decryption result. Next, it encrypts the addition result with a Paillier public key ppk_j ($2ModExp + ModMul$), and then encrypts psk with CP-ABE using AP_M ($(|\gamma|+1)Exp$). Hence, the computation cost of CP in the DRs-DOs case is: $(N_m + 2)ModExp + (N_m + 1)ModMul + (|\gamma| + 1)Exp$. Therefore, in total, the computational complexity of both in the DRs-DO case is: $(3N_m + 4)ModExp + (3N_m + 3)ModMul + (|\gamma| + 1)Exp$.

Computations at DRs: For the DO-DO case, a DR (who is the DO in practice) uses its wsk_i to decrypt the processing results ($ModExp + ModMul$). Whereas, for DRs-DO and DRs-DOs use cases, a DR decrypts the ABE ciphertext using their sk to obtain the Paillier decryption key psk_j (at most $\vartheta BiPair$) and then uses it to decrypt the encrypted processing result ($ModExp + ModMul$). This gives a computational cost at DR: ($ModExp + ModMul + \vartheta BiPair$).

We compare the total computational costs of each entity in SAMA with the addition scheme of [20] and [28] in Table V.

TABLE V: Computation Cost.

	SAMA: DO-DO case
DO	$2ModExp + ModMul$
SP	$(N_m - 1)ModMul$
CP	$--$
DO^*	$ModExp + ModMul$
	SAMA: DRs-DO case
DO	$2ModExp + ModMul$
SP	$4ModExp + (N_m + 3)ModMul$
CP	$3ModExp + 2ModMul + (|\gamma| + 1)Exp$
DR	$ModExp + ModMul + \vartheta BiPair$
	SAMA: DRs-DOs case
DO	$2ModExp + ModMul$
SP	$(2N_m + 2)ModExp + (2N_m + 2)ModMul$
CP	$(N_m + 2)ModExp + (N_m + 1)ModMul + (|\gamma| + 1)Exp$
DR	$ModExp + ModMul + \vartheta BiPair$
	Scheme in [20]: DRs-DOs case (ADD operation)
DO	$2ModExp + ModMul$
SP	$3ModExp + (N_m + 2)ModMul$
CP	$9ModExp + (N_m + 4)ModMul + 2(|\gamma| + 1)Exp$
DR	$ModExp + ModMul + ModInverse + \vartheta BiPair$
	Scheme in [28]: DRs-DOs case
DO	$3ModExp + ModMul$
SP	$(2N_m + 2)ModExp + (3N_m + 2)ModMul$
CP	$9ModExp + (2N_m + 4)ModMul + 2(|\gamma| + 1)Exp$
DR	$ModExp + ModMul + ModInverse + \vartheta BiPair$

* DO acts as DR; All modular exponentiation and modular multiplications modulo are under n^2

2) Computational Complexity of Access Control: We assume there are $|U|$ universal attributes, in which $|\gamma|$ attributes are in the access policy tree τ, and at most ϑ attributes should be satisfied in τ to decrypt the ciphertext. The $Setup()$ will generate the public parameters using the given system parameters and attributes U. This requires $|U| + 1$ exponentiations and one bi-linear pairing. The $\mathsf{Enc}_{ABE}()$ requires two exponential operations for each leaf in the ciphertext's access tree τ, which needs $(|\gamma| + 1)Exp$. In contrast, the $\mathsf{KGen}_{ABE}()$ algorithm requires two exponential operations for every attribute given to the data owner. Also, the private key consists of two group elements for every

TABLE VI: Communication Overhead.

	SAMA: DO-DO case
DO-to-SP	$2NL(n)$
SP-to-CP	–
SP-to-DO	$2L(n)$
	SAMA: DRs-DO case
DOs-to-SP	$2NL(n)$
SP-to-CP	$4L(n) + (\|\gamma\| + 1)\mathcal{L}$
SP-to-DR	$2L(n) + (\|\gamma\| + 1)\mathcal{L}$
	SAMA: DRs-DOs case
DOs-to-SP	$2NL(n)$
SP-to-CP	$2(N + 1)L(n) + (\|\gamma\| + 1)\mathcal{L}$
SP-to-DR	$2L(n) + (\|\gamma\| + 1)\mathcal{L}$
	Scheme in [20]: DRs-DOs case (ADD)
DOs-to-SP	$4NL(n)$
SP-to-CP	$8L(n) + (\|\gamma\| + 1)\mathcal{L}$
SP-to-DR	$4L(n) + (\|\gamma\| + 1)\mathcal{L}$
	Scheme in [28]: DRs-DOs case
DOs-to-SP	$4NL(n) + \|\sigma\|$
SP-to-CP	$8L(n) + (\|\gamma\| + 1)\mathcal{L}$
SP-to-DR	$4L(n) + (\|\gamma\| + 1)\mathcal{L}$

attribute. Finally, $\mathsf{Dec}_{ABE}()$ requires two pairings for every leaf of the access tree τ matched by a private key attribute and, at most, one exponentiation for each node along a path from that leaf to the root node.

B. Communication Overhead

There are two types of communication overhead incurred in the SAMA scheme: overhead due to occasional data communication and overhead due to regular data communication. The former overhead captures the data sent occasionally, e.g., AP (AP_S, AP_M) uploads/updates and notifications. The latter overhead includes the regular data communication patterns within SAMA, such as data upload, data requests, and data exchanged between cloud providers when data is being processed. Since the former overhead is negligible compared to the latter, we focus only on the communication overhead due to regular data communication patterns.

To ease the analyses, we divide the communication overhead introduced by the SAMA scheme into three parts: overhead incurred (1) between DOs and SP denoted as (DOs-to-SP), (2) between SP and CP (SP-to-CP), and (3) between SP and DRs (SP-to-DRs).

1) DOs-to-SP: This is a common step for the DO-DO, DRs-DO and DRs-DOs use cases. Each data owner DO_i sends one ciphertext to SP at each data reporting time slot. As each ciphertext has a length of $2L(n)$ (operations are performed under $mod\, n^2$), the total communication overhead for this part in the DRs-DO and DRs-DOs use cases is $N2L(n)$.

2) SP-to-CP: The communication between SP-to-CP applies only to the DRs-DO and DRs-DOs cases, and it is as follows. For the DRs-DO, SP sends one ciphertext of length $2L(n)$, which is the masked aggregated data owner's data, to the CP. Then, the CP sends one ciphertext of $2L(n)$ to the SP, which is the masked encrypted processing result, and one CP-ABE ciphertext of $(|\gamma|+1)\mathcal{L}$, where $\mathcal{L}$ is the bit length of elements in ABE. Therefore, the total communication among SP-to-CP in the DRs-DO case is: $4L(n)+(|\gamma|+1)\mathcal{L}$.

The communication between SP-to-CP in the DRs-DOs is as follows. SP sends N ciphertext (masked of encrypted DO's data) of length $2L(n)$ to the CP, which is $2NL(n)$. Then, similar to the DRs-DO scenario, the CP sends one ciphertext of $2L(n)$ and $(|\gamma|+1)\mathcal{L}$ of the CP-ABE ciphertext to the SP. The total communication cost among SP-to-CP in the DRs-DOs case is: $(N + 1)2L(n) + (|\gamma| + 1)\mathcal{L}$.

3) SP-to-DRs: For the DO-DO, SP sends one ciphertext of length $2L(n)$ (the encrypted processing result) to the DO(DR). Thus, the communication overhead between SP-to-DRs in the DO-DO case is: $2L(n)$ Whereas, in the DRs-DO and DRs-DOs, the SP send an extra one CP-ABE ciphertext of length $(|\gamma| + 1)\mathcal{L}$. Thus, The communication between SP-to-DRs in the DRs-DO and DRs-DOs cases is: $2L(n) + (|\gamma| + 1)\mathcal{L}$.

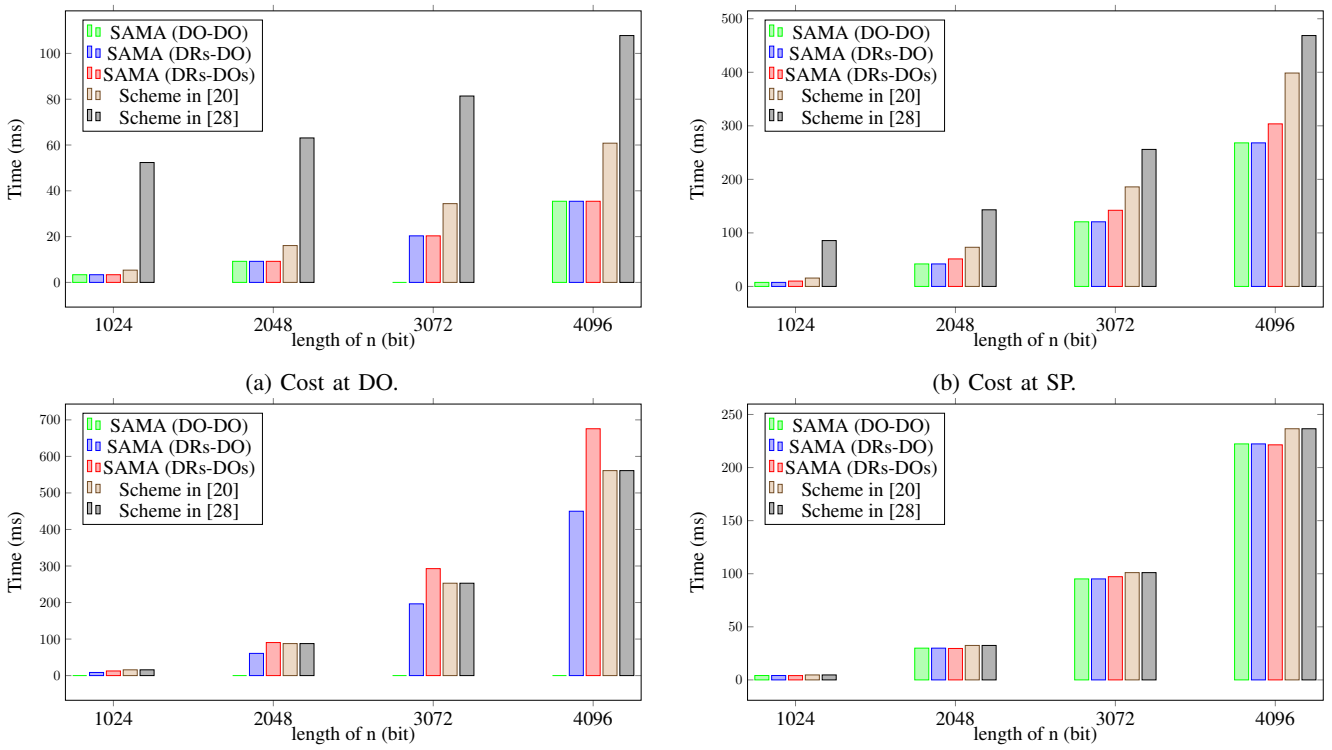
A comparison between the communication overhead of the SAMA scheme and the addition schemes proposed in [20] and [28] is shown in Table VI. Overall, SAMA has a lower communication overhead than both schemes at the DO and DR side, while the overhead between SP-to-CP in the DRs-DOs of the SAMA scheme is higher than the overhead in [20] and [28].

C. Experimental Results

We present the experimental results of SAMA in three different settings: (1) computational cost of the data processing operations, (2) computational cost of the data access operations, and (3) communication overheads within SAMA.

For the computational cost, we have implemented the SAMA scheme to test its computational performances by conducting experiments with Java Pairing-Based Cryptography (jPBC) [43] and Java Realization for Ciphertext-Policy Attribute-Based Encryption (cpabe) [44] libraries on a laptop with Intel Core i7-7660U CPU 2.50GHz and 8GB RAM. We ran each experiment 500 times and took the average values. We set the length of n to 1024 bits, m to 250 bits, and r to 500 bits according to [20] to make the comparison compatible. We show the computation evaluation for the DRs-DO and DRs-DOs use cases for all entities separately and specifically SP and CP, as they perform different sets of computations in each case as described in Section VI-C1. In addition, the efficiency of user-centric access control and communication overhead among the entities are shown in Section VI-C2 and Section VI-C3, respectively.

1) Computational Cost of Data Processing: We evaluate the computational cost for DO, SP, CP, and DR in the DRs-DO and DRs-DOs scenarios and compare with the related



(a) Cost at DO.

(b) Cost at SP.

(c) Cost at CP.

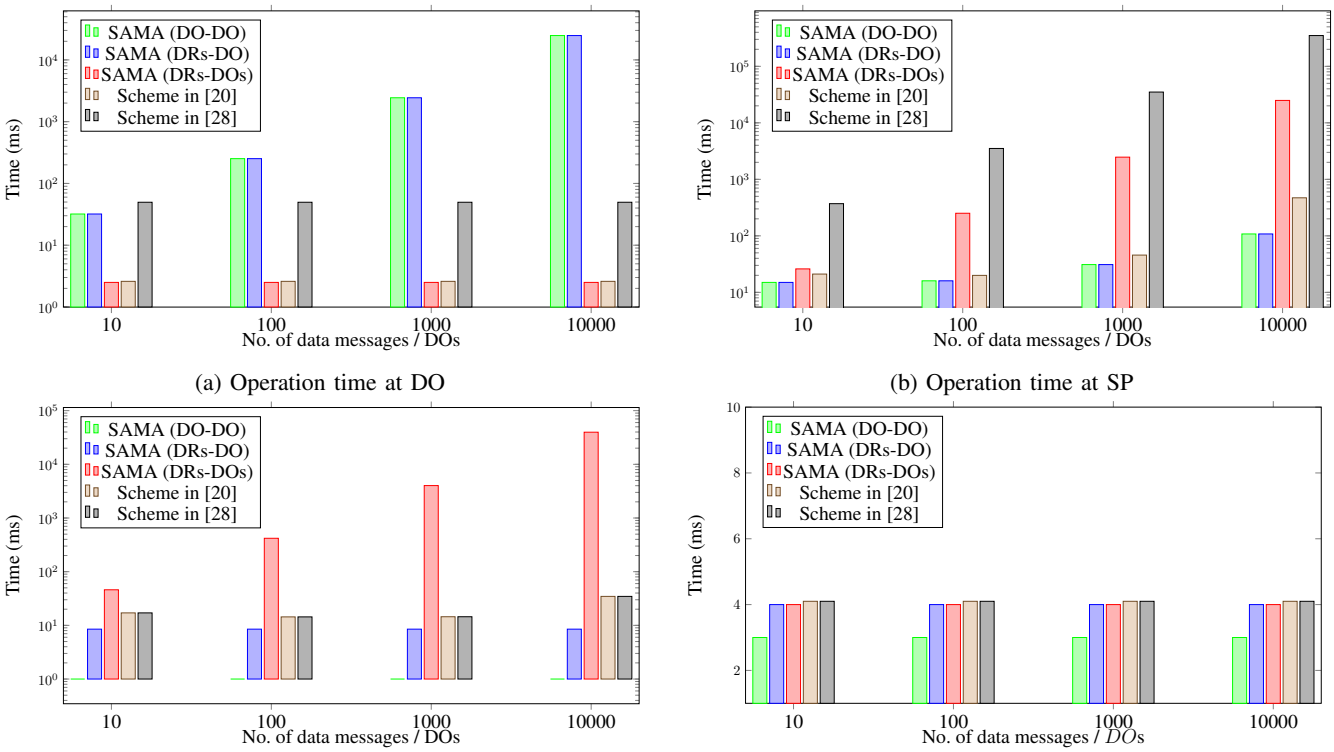
(d) Cost at DR.

Fig. 4: SAMA scheme cost with the different lengths of n.

works [20], [28] (DRs-DOs use case) in terms of different lengths of n. For the DO-DO case, we only evaluate the computational cost for DO and SP as the data is only processed at SP and shared with the DO. In addition, we show the computational cost of the DO-DO, DRs-DO and DRs-DOs cases with a variable number of messages and DOs, respectively.

(i) Influence of different lengths of n on data processing: Figure 4 shows the influence of the different lengths of n on data processing of two messages, where n=1024, 2048, 3072, and 4096 bits. We can observe that in Fig. 4a, the computational cost of SAMA is low on the data owner side and the lowest among all the other entities because the data owner only needs to encrypt data once, which is suitable for resource-constrained devices. Note, although DOs encrypt their data once to support the DO-DO, DRs-DO and DRs-DOs cases, we show three separate bars for consistency. Moreover, since in the encryption, there is an extra addition and multiplication that depends on n (key size) in [20], [28] compared to SAMA, the experimental results show that SAMA's data owner side encryption is better than [20], [28]. In our DO-DO, DRs-DO and DRs-DOs cases, SP achieves better computational efficiency compared to the DSP scheme in [20], [28], as shown in Fig. 4b. The computational efficiency of CP in our DRs-DO is better than the CP of the scheme in [20], [28] as shown in Fig. 4c. Whereas the computational efficiency of CP is slightly lower in our DRs-DOs compared to the CP of the scheme in [20], [28]. The operation time of the DR, as shown in Fig. 4d is marginally better than the scheme in [20], [28], since the decryption of [20], [28] can not be optimised by pre-computation as the decryption is dependent on ciphertexts. Therefore, there is an extra $modInverse$ operation compared to SAMA. Whereas in SAMA, the denominator of decryption needs to be computed only once and is not dependent on the ciphertext; hence it can be pre-computed.

We can observe that the computation cost is linearly increasing with the increase of the bit length of n among all entities: DO, SP, CP, and DR. However, as expected, SP and CP computation costs increase much more rapidly with the increase of bit length of n compared to the DO and DR in the cases of DO-DO and DRs-DOs. The computational performance evaluation shown in Fig. 4 is consistent with our analysis in Section VI-A1. In general, the above tests prove that the most computation costs are undertaken at SP and CP and DO/DR do not have much computation overhead. This result shows the practical advantage of SAMA with DOs and DRs. Also, overall, our scheme performs better in terms of efficiency compared to the schemes in [20], [28] which supports only the DRs-DOs case.



(a) Operation time at DO

(b) Operation time at SP

(c) Operation time at CP

(d) Operation time at DR

Fig. 5: Cost with different numbers of messages/DOs.

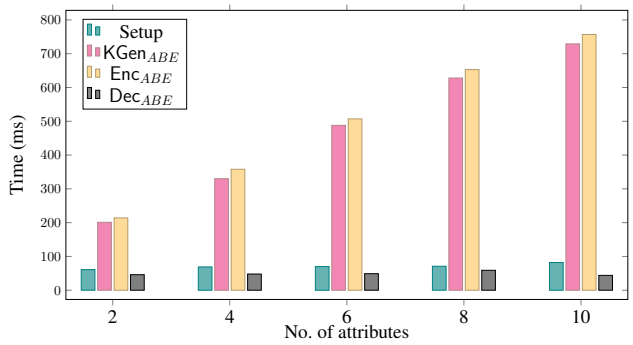
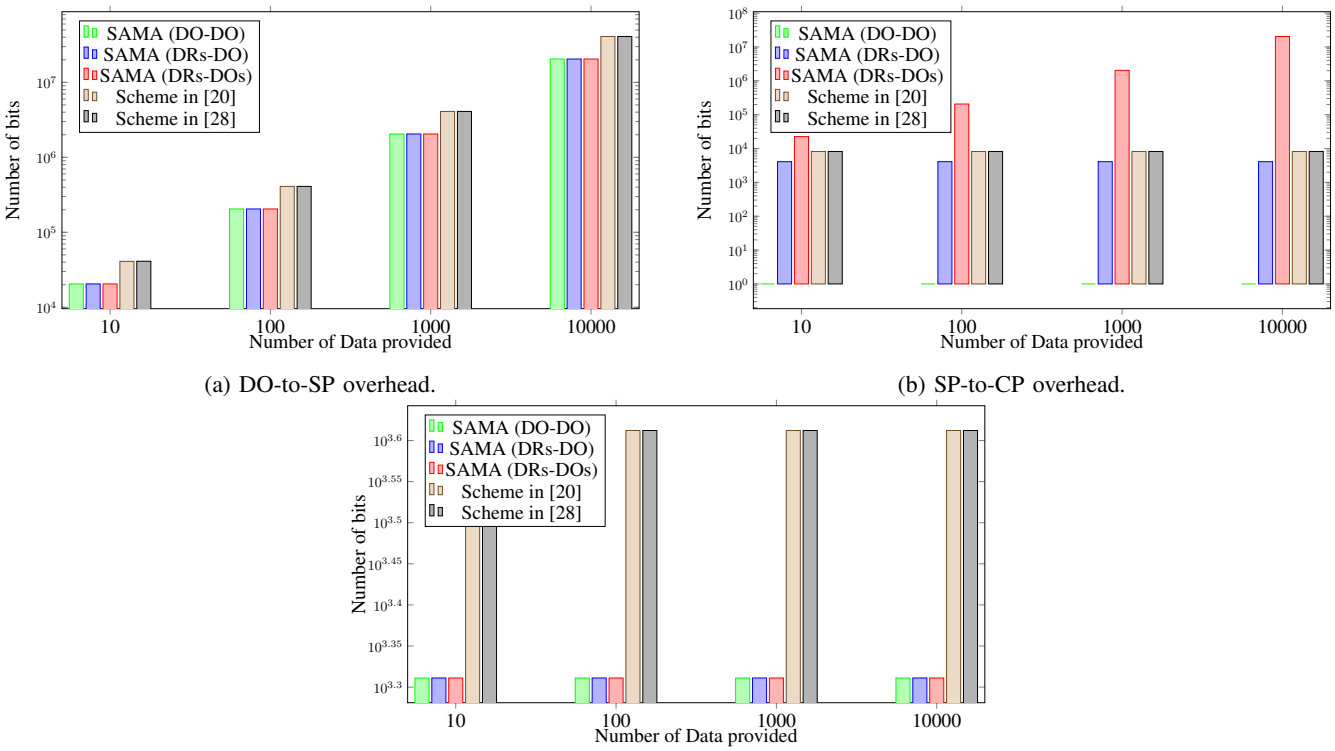


Fig. 6: CP-ABE time with the different numbers of attributes.

(ii) Performance of SAMA (DRs-DO case) with a variable number of provided messages: We tested the computation of SAMA's DO-DO and DRs-DO cases by varying the number of data messages provided by a single DO, as shown in Fig. 5. As seen from Fig. 5a, the operational time of DO and SP increases with the increase in the number of messages. However, only the DR's and CP's operation times are independent of the number of messages because, in the DRs-DO case, the aggregated result is decrypted only once, regardless of the number of messages processed at the SP. This is similar to the operational time of the DR, who acts as the DO in the DO-DO case.

(iii) Performance of SAMA (DRs-DOs case) with a variable number of data owners: We tested the performance of SAMA's DRs-DOs case by varying the number of DOs (N_{DO}= 10, 100, 1000, 10000) and fixing each data owner to generate only one message for data processing. As expected, the SP and CP have more operation time compared to the DO and DR. Moreover, as shown in Fig. 5, the SP and CP operation time are higher in the DRs-DOs case compared to the [20] and DRs-DO case. Since VP-HE is similar to PE used in [20], [28] that supports only single-key homomorphic addition, our DRs-DO case processing is comparable to the [20] and our DRs-DOs computation time is higher only at the cloud side (SP and CP) due to separate masking and de-masking for each of the messages.

2) Efficiency of User-Centric Access Control: We tested the computational efficiency of CP-ABE by varying the number of attributes from two to ten that are involved in the access policy, as shown in Fig. 6. The $Setup$ algorithm is relatively constant as it does not depend on the number of attributes. In addition, the Dec_{ABE} in the test was set to require only one attribute to satisfy the access policy tree. Therefore, the operation time of Dec_{ABE} is constant. The computational costs of Enc_{ABE} and KGen_{ABE} are linearly increasing with the increase in the number of attributes.



(a) DO-to-SP overhead.

(b) SP-to-CP overhead.

(c) SP-to-DR overhead.

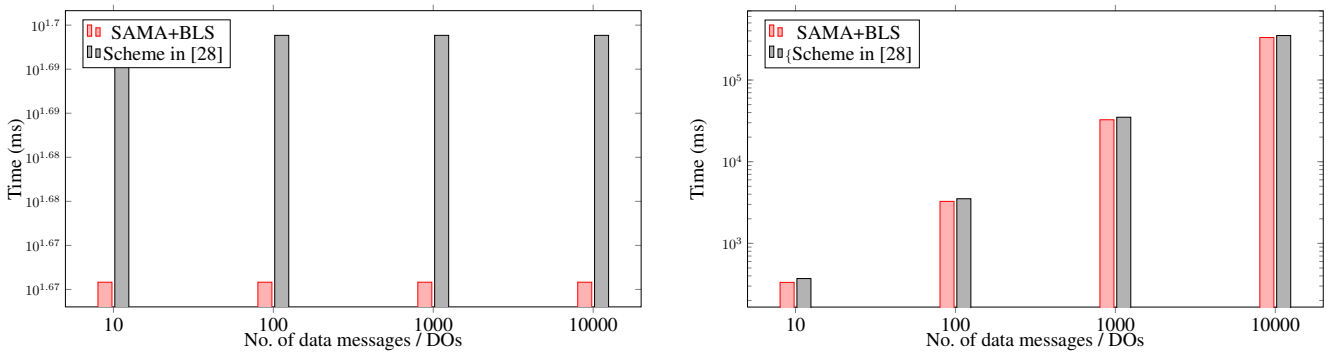
Fig. 7: Communication overhead of SAMA.

Although employing CP-ABE achieves user-centric fine-grain access control, an additional computation overhead is incurred.

3) Communication Efficiency: The communication overhead among the entities is shown in Fig. 7, and it is evaluated by fixing the key size length $n = 1024$ bits and varying the number of messages to be computed. It is evident from Fig. 7a, the DO-to-SP communications at the SAMA scheme reduce the communication overhead by 50% compared to the scheme in [20], [28]. Furthermore, it is essential to note that the schemes in [20], [28] support only the DRs-DOs case by encrypting data with CSP's public key. To support the DO-DO case, they need to re-encrypt the same data again with the DO's public key as mentioned in [20]. Therefore, if we also compare the DRs-DO communication overhead of the scheme in [20] at the DO side, our scheme reduces the communication overhead by 75%. At SAMA, a DO has to encrypt wearable data only once for DO-DO, DRs-DO and DRs-DOs cases compared to the scheme in [20], [28] which requires encrypting the data owner's data twice to support both DRs-DO and DRs-DOs. In addition, the schemes in [20], [28] generate two ciphertexts for every data encryption, which increases communication overhead on the DO side. While in the SAMA scheme, only one ciphertext is generated. Furthermore, the scheme in [28] has slightly more communication overhead than [20] at DO due to the signatures. Clearly, we reduced the communication overhead significantly on the DO side, which suits the resource-constrained devices. These results are consistent with the results obtained in [10], which compares the communication overhead of the two HE algorithms: BCP and VP-HE. They found that the communication cost of BCP is about twice that of VP-HE, which was used in [20], [28].

Figure 7b depicts the communication overhead among the cloud servers (SP-to-CP and CP-to-SP). Although our DRs-DO case achieves better communication efficiency compared to [20], [28], our DRs-DOs case communication performance is significantly higher than the DRs-DOs case of [20], [28]. However, since SP and CP are not limited in resources, they can afford to support this higher communication overhead for the DRs-DOs case. Moreover, the frequency of DRs-DOs cases is relatively less than DO-DO and DRs-DO cases in most wearable and healthcare use cases that are more personalized. We achieve better communication efficiency with the most frequent DRs-DO cases. Therefore, our scheme is suitable mainly for applications that require more frequent DO-DO and DRs-DO cases than DRs-DOs cases such as wearables and outsourced personalized healthcare data processing.

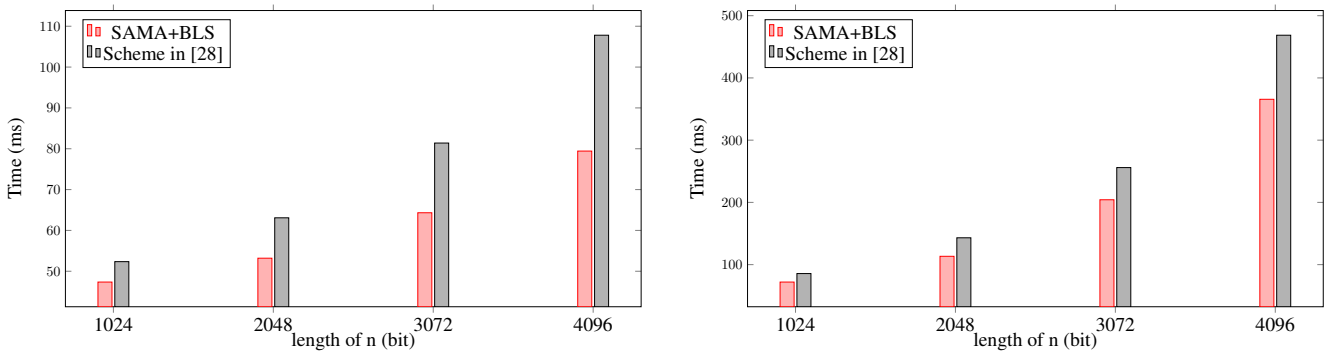
The communication overhead of the SP-to-DR part is



(a) Operation time at DO

(b) Operation time at SP

Fig. 8: Cost with different numbers of messages/DOs.



(a) Cost at DO.

(b) Cost at SP.

Fig. 9: SAMA scheme cost with the different lengths of n.

shown in Fig. 7c. As DRs access only the processed result, there is less communication overhead between the SP and DR. It is clear that our DRs-DO and DRs-DOs cases perform better than the scheme in [20], [28] which supports only DRs-DOs case. Therefore, overall our scheme has significantly less total communication overhead compared to [20] and [28].

4) Comparison with [28]: As already mentioned, the solution [28] builds upon the approach proposed in [20] with a signature technique applied on top of it to support integrity protection. We have added the work [28] to our evaluation. As shown in Fig. 4, 5, and 7, the evaluation results demonstrate that [28] has higher or similar computation costs to [20] in all cases except for the DO cost (see Fig. 5a). Note: the BLS signature generation and verification processes occur only at DO and SP. For a fair comparison, we have also added signature generation and verification steps to SAMA (denoted as SAMA+BLS), even though it is not the main focus of our research. In Fig. 8 and 9, the evaluation results reveal that SAMA performs better than [28], even with the added signature.

VII. Conclusion

In this paper, we have designed and evaluated a novel flexible data processing with a fine-grain sharing scheme called SAMA that also supports user-centric access control. It addresses the diverse demand for modern wearable healthcare applications by accommodating all three main use-case scenarios: DO-DO, DRs-DOs, and DRs-DOs. To achieve this, SAMA combines multi-key VP-HE and CP-ABE to support efficient and privacy-preserving aggregation with fine-grain sharing. In addition, SAMA is suitable for resource-constrained devices like wearable devices as we ensure that data owners encrypt their data only once with their own public key, yet SAMA still supports all three use cases. Our experimental evaluation and comparison with the closest previous works [20], [28] demonstrate that SAMA is more efficient in terms of computation and communication cost. Further, the resource-intensive CP-ABE is outsourced to the cloud, reducing the burden on data owners and yet allowing them to set two different access policies to control their data and thereby achieve user-centric access control. Our security analysis through the simulation paradigm shows that SAMA is secure and fulfils the specified set of security

and privacy requirements.

REFERENCES

[1] V. G. Motti, "Wearable interaction," in *Wearable Interaction*. Springer, 2020, pp. 81–107.

[2] M. M. Islam, S. Mahmud, L. Muhammad, M. R. Islam, S. Nooruddin, and S. I. Ayon, "Wearable technology to assist the patients infected with coronavirus (covid-19)," *SN Computer Science*, vol. 1, no. 6, pp. 1–9, 2020.

[3] J. Zhou, Z. Cao, X. Dong, and X. Lin, "Security and privacy in cloud-assisted wireless wearable communications: challenges, solutions, and future directions," *IEEE Wireless Communications*, vol. 22, no. 2, pp. 136–144, 2015.

[4] H. Shafagh, L. Burkhalter, S. Ratnasamy, and A. Hithnawi, "Droplet: Decentralized authorization and access control for encrypted data streams," in *USENIX Security Symposium*, 2020, pp. 2469–2486.

[5] F. Wang, J. Mickens, N. Zeldovich, and V. Vaikuntanathan, "Sieve: Cryptographically enforced access control for user data in untrusted clouds," in *USENIX Symposium on Networked Systems Design and Implementation*, 2016, pp. 611–626.

[6] M. Li, S. Yu, Y. Zheng, K. Ren, and W. Lou, "Scalable and secure sharing of personal health records in cloud computing using attribute-based encryption," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 131–143, 2012.

[7] "Regulation (EU) 2016 - general data protection regulation," (Accessed on 02/03/2022). [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32016R0679

[8] A. Act, "Health insurance portability and accountability act of 1996," *Public Law*, vol. 104, p. 191, 1996.

[9] S. Safavi and Z. Shukur, "Conceptual privacy framework for health information on wearable device," *PLoS ONE*, vol. 9, no. 12, pp. 1–16, 2014.

[10] H. Pang and B. Wang, "Privacy-preserving association rule mining using homomorphic encryption in a multikey environment," *IEEE Systems Journal*, vol. 15, no. 2, pp. 3131–3141, 2021.

[11] W. Ding, Z. Yan, and R. H. Deng, "Encrypted data processing with homomorphic re-encryption," *Information Sciences*, vol. 409, pp. 35–55, 2017.

[12] A. Ara, M. Al-Rodhaan, Y. Tian, and A. Al-Dhelaan, "A secure privacy-preserving data aggregation scheme based on bilinear elgamal cryptosystem for remote health monitoring systems," *IEEE Access*, vol. 5, pp. 12 601–12 617, 2017.

[13] G. Wang, R. Lu, and Y. L. Guan, "Achieve privacy-preserving priority classification on patient health data in remote ehealthcare system," *IEEE Access*, vol. 7, pp. 33 565–33 576, 2019.

[14] Z. Erkin, T. Veugen, T. Toft, and R. L. Lagendijk, "Generating private recommendations efficiently using homomorphic encryption and data packing," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1053–1066, 2012.

[15] H. Li, Q. Cheng, X. Li, S. Ma, and J. Ma, "Lightweight and fine-grained privacy-preserving data aggregation scheme in edge computing," *IEEE Systems Journal*, vol. 16, no. 2, pp. 1832–1841, 2022.

[16] Y. Zhang, W. Dai, X. Jiang, H. Xiong, and S. Wang, "Foresee: Fully outsourced secure genome study based on homomorphic encryption," in *BMC medical informatics and decision making*, vol. 15, no. 5. Springer, 2015, pp. 1–11.

[17] A. Aloufi, P. Hu, Y. Song, and K. Lauter, "Computing blindfolded on data homomorphically encrypted under multiple keys: A survey," *ACM Computing Surveys*, vol. 54, no. 9, pp. 1–37, 2021.

[18] J. Zhang, Z. L. Jiang, P. Li, and S. M. Yiu, "Privacy-preserving multikey computing framework for encrypted data in the cloud," *Information Sciences*, vol. 575, pp. 217–230, 2021.

[19] B. Wang, M. Li, S. Chow, and H. Li, "A tale of two clouds: Computing on data encrypted under multiple keys," in *2014 IEEE Conference on Communications and Network Security*, 2014, pp. 337–345.

[20] W. Ding, Z. Yan, and R. Deng, "Privacy-preserving data processing with flexible access control," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 2, pp. 363–376, 2020.

[21] M. A. Mustafa, N. Zhang, G. Kalogridis, and Z. Fan, "Dep2sa: A decentralized efficient privacy-preserving and selective aggregation scheme in ami," *IEEE Access*, vol. 3, pp. 2828–2846, 2015.

[22] A. López-Alt, E. Tromer, and V. Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption," in *ACM symposium on Theory of computing*, 2012, pp. 1219–1234.

[23] J. Chen, Y. Feng, Y. Liu, W. Wu, and G. Yang, "Non-interactive privacy-preserving naïve bayes classifier using homomorphic encryption," in *International Conference on Security and Privacy in New Computing Environments*. Springer, 2021, pp. 192–203.

[24] A. Aloufi, P. Hu, H. W. Wong, and S. S. Chow, "Blindfolded evaluation of random forests with multi-key homomorphic encryption," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 4, pp. 1821–1835, 2019.

[25] H. Li, K. Yu, B. Liu, C. Feng, Z. Qin, and G. Srivastava, "An efficient ciphertext-policy weighted attribute-based encryption for the internet of health things," *IEEE Journal of Biomedical and Health Informatics*, 2021.

[26] S. Alshehri, S. P. Radziszowski, and R. K. Raj, "Secure access for healthcare data in the cloud using ciphertext-policy attribute-based encryption," in *IEEE 28th international conference on data engineering workshops*. IEEE, 2012, pp. 143–146.

[27] S. Narayan, M. Gagné, and R. Safavi-Naini, "Privacy preserving ehr system using attribute-based infrastructure," in *Proceedings of the 2nd ACM Cloud Computing Security Workshop*, 2010, pp. 47–52.

[28] S. Chen, J. Wang, Z. Yu, H. Xu, and C. Dong, "A cloud-assisted data processing scheme for smart grid with flexible access control," in *Proceedings of the 2022 5th International Conference on Algorithms, Computing and Artificial Intelligence*, 2022, pp. 1–6.

[29] J. Wang, H. Feng, Z. Yu, R. Liao, S. Chen, and T. Liang, "Secure and efficient data processing for cloud computing with fine-grained access control," in *International Symposium on Security and Privacy in Social Networks and Big Data*. Springer, 2023, pp. 187–202.

[30] S. Ruj and A. Nayak, "A decentralized security framework for data aggregation and access control in smart grids," *IEEE Transaction on Smart Grid*, vol. 4, no. 1, pp. 196–205, 2013.

[31] T. Bhowmik and I. Banerjee, "Eeppda—edge-enabled efficient privacy-preserving data aggregation in smart healthcare internet of things network," *International Journal of Network Management*, p. e2216, 2023.

[32] W. Zhang, S. Liu, and Z. Xia, "A distributed privacy-preserving data aggregation scheme for smart grid with fine-grained access control," *Journal of Information Security and Applications*, vol. 66, 2022.

[33] W. Tang, J. Ren, K. Zhang, D. Zhang, Y. Zhang, and X. Shen, "Efficient and privacy-preserving fog-assisted health data sharing scheme," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 6, pp. 1–23, 2019.

[34] L. Liu, H. Wang, and Y. Zhang, "Secure iot data outsourcing with aggregate statistics and fine-grained access control," *IEEE Access*, vol. 8, pp. 95 057–95 067, 2019.

[35] H. Wang, J. Liang, Y. Ding, S. Tang, and Y. Wang, "Ciphertext-policy attribute-based encryption supporting policy-hiding and cloud auditing in smart health," *Computer Standards & Interfaces*, vol. 84, p. 103696, 2023.

[36] A. Sahai and B. Waters, "Fuzzy identity-based encryption," in *Int.l Conf. on the TACT*. Springer, 2005, pp. 457–473.

[37] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *2007 IEEE symposium on security and privacy*, 2007, pp. 321–334.

[38] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*, 2006, pp. 89–98.

[39] W. Ding, R. Hu, Z. Yan, X. Qian, R. H. Deng, L. T. Yang, and M. Dong, "An extended framework of privacy-preserving computation with flexible access control," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 918–930, 2020.

[40] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International conference on the theory and*

applications of cryptographic techniques. Springer, 1999, pp. 223–238.

[41] Y. Lindell, "How to simulate it–a tutorial on the simulation proof technique," *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, pp. 277–346, 2017.

[42] J. Liu, X. Huang, and J. K. Liu, "Secure sharing of personal health records in cloud computing: Ciphertext-policy attribute-based signcryption," *Future Generation Computer Systems*, vol. 52, pp. 67–76, 2015.

[43] A. De Caro and V. Iovino, "jpbc: Java pairing based cryptography," in *Symposium on Computers and Communications*. IEEE, 2011, pp. 850–855. [Online]. Available: http://gas.dia.unisa.it/projects/jpbc/

[44] J. Wang, "Java realization for ciphertext-policy attribute-based encryption," 2012. [Online]. Available: https://github.com/junwei-wang/cpabe/