Securely Offloading Computation to the Edge with the Tangle Cache

Aqsa Ayub[1]*, Muhammad Rizwan[2], Shivanshu Shrivastava[3], Adeel Anjum[4], Pelin Angin[5], and Yigit Sever[5]

[1]Department of Computer Science, COMSATS University Islamabad, Islamabad, Pakistan
aqsaayub.21@gmail.com

[2]University of Science and Technology of China
rizwanramay@gmail.com

[3]Department of Electronics Engineering, Rajiv Gandhi Institute of Petroleum Technology, Amethi-229304, India
sshrivastava@rgipt.ac.in

[4]Quaid-i-Azam University Islamabad, Islamabad, Pakistan
aanjum@qau.edu.pk

[5]Department of Computer Engineering, Middle East Technical University, Ankara, Turkey
{pangin, yigit}@ceng.metu.edu.tr

Received: September 30, 2022; Accepted: December 4, 2022; Published: December 31, 2022

Abstract

The rising area of the Internet of Things (IoT) faces various issues related to the scalability and security of the data produced. Most IoT devices cannot perform complex computations on their own due to resource constraints, which require them to outsource their jobs. Due to the abundance of devices and the IoT network's heterogeneous nature, there is a large attack surface against IoT systems. During outsourcing jobs, a major factor that affects performance is the delay in data transmission. However, the security issues such as data manipulation and leakage for edge caching have not been fully addressed in existing literature. This paper proposes a Tangle based caching model for edge services, as well as an efficient and effective scheme that will aid the offloading process between IoT devices and edge devices. The proposed model utilizes reversible integer transformation in the context of edge computing to build a secure data transfer technique that distorts the original data so that an adversary cannot access it. Experimental studies and theoretical evaluation demonstrate that our proposed scheme is promising for adoption in future IoT systems with its efficiency and security features.

Keywords: Blockchain, Edge Computing, Edge cache service, IoT, Tangle

1 Introduction

The rapid advances in the Internet technologies in recent years have paved the way for the future Internet of Things (IoT), due to the billions of new devices which will become part of the globally connected

Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 13(4):40-60, Dec. 2022
DOI:*10.58346/JOWUA.2022.I4.003*

*Corresponding author: Department of Computer Science, COMSATS University Islamabad, Islamabad, Pakistan
Email: aqsaayub.21@gmail.com

network every year. According to recent estimates, [1], over 50 billion devices will be sharing the network by 2025. With this rapid rise in the total number of IoT devices worldwide, the volume of data produced by these devices has also been increasing significantly and will continue to grow in the future, bringing along challenges such as the need for mechanisms that can be used to make the best use of the collected data. IoT also faces major issues related to the security and scalability of the produced data and the underlying networks. Most IoT devices are resource-constrained, hence they rely on other resource-rich nodes to perform complex tasks, which leads to data privacy and security concerns. Using a traditional centralized cloud environment poses various challenges for data privacy since the data has to be shared with a third-party service provider. Moreover, a cloud-based IoT architecture will face performance issues due to the relatively long distance between the devices and the remote data centers, creating delays in the communication of data between the involved nodes. To enhance the performance and scalability of the network, computation needs to be performed closer to the IoT devices.

The forthcoming communications infrastructure is envisioned as a large-scale, densely linked sophisticated mobile network capable of growing an intelligent social system that can adapt to surges in user needs regardless of time or location. As a result of the expected future prerequisites for 6G, mobile edge computing (MEC), which is the solution to the problem of decentralized computation offloading in IoT, will undoubtedly evolve into a technology that is assisted by artificial intelligence (AI), capable of providing smart services to mobile devices via computationally efficient hardware architecture, effective caching technology, and AI algorithms [2, 3]. By offloading computation to servers physically close to IoT devices, edge computing improves the scalability of the network and solves the problem of managing a large number of IoT devices by addressing their computing, storage, and intelligent data processing needs. The infrastructure of edge computing is given in the Fig.1.

As mentioned above, IoT devices are vulnerable to various attacks due to being connected to a large network with a large attack surface. Complex security protocols cannot be deployed on most IoT devices due to their resource-constrained nature. In order to exploit these vulnerabilities, adversaries operate botnets to perform various attacks [4]. The connectivity of compromised IoT devices can have adverse consequences for other devices connected to the network as adversaries can easily penetrate the network of the IoT devices through compromised devices and move laterally. The main reason behind these classes of vulnerabilities is that cheap and low-cost devices have lower security and seldom have technical upgrades or maintenance services. To protect against this increased vulnerability of IoT devices, some techniques have been proposed [5]. Since data owners are concerned about the confidential information in their data being exposed without authorization in an edge computing ecosystem, they strive to use privacy-preserving strategies on the data before publishing it. Even in architectures where the data collectors would not be able to get the full data from the users, the users would still want their data to be secured before exiting their device.

We need to take extra security measures to protect the data handled by the IoT devices, since outdated general-purpose security solutions cannot provide adequate security for them. Due to the nature of the IoT devices, lightweight security mechanisms need to be deployed to provide security with lower resource and computation requirements. Blockchain [6, 7, 8, 9, 10] is an emerging technology that has gained the interest of researchers due to the services it provides for the integrity of data shared in a large network. A decentralized computing environment based on blockchain has the potential to alleviate the computation and resource utilization problems of IoT while providing strong security for the network.

Despite providing increased availability of data for transactions, traditional blockchains face some issues regarding scalability and efficiency. For instance, the proof-of-work (PoW) consensus protocol requires a large number of devices utilizing their resources to participate in the network. The expenditure of the computation power to move the chain forward is designed to keep the participants honest to the network with a trade-off for scalability and efficiency.

In this paper, we propose a secure cache-aware scheme to aid the process of computation offloading

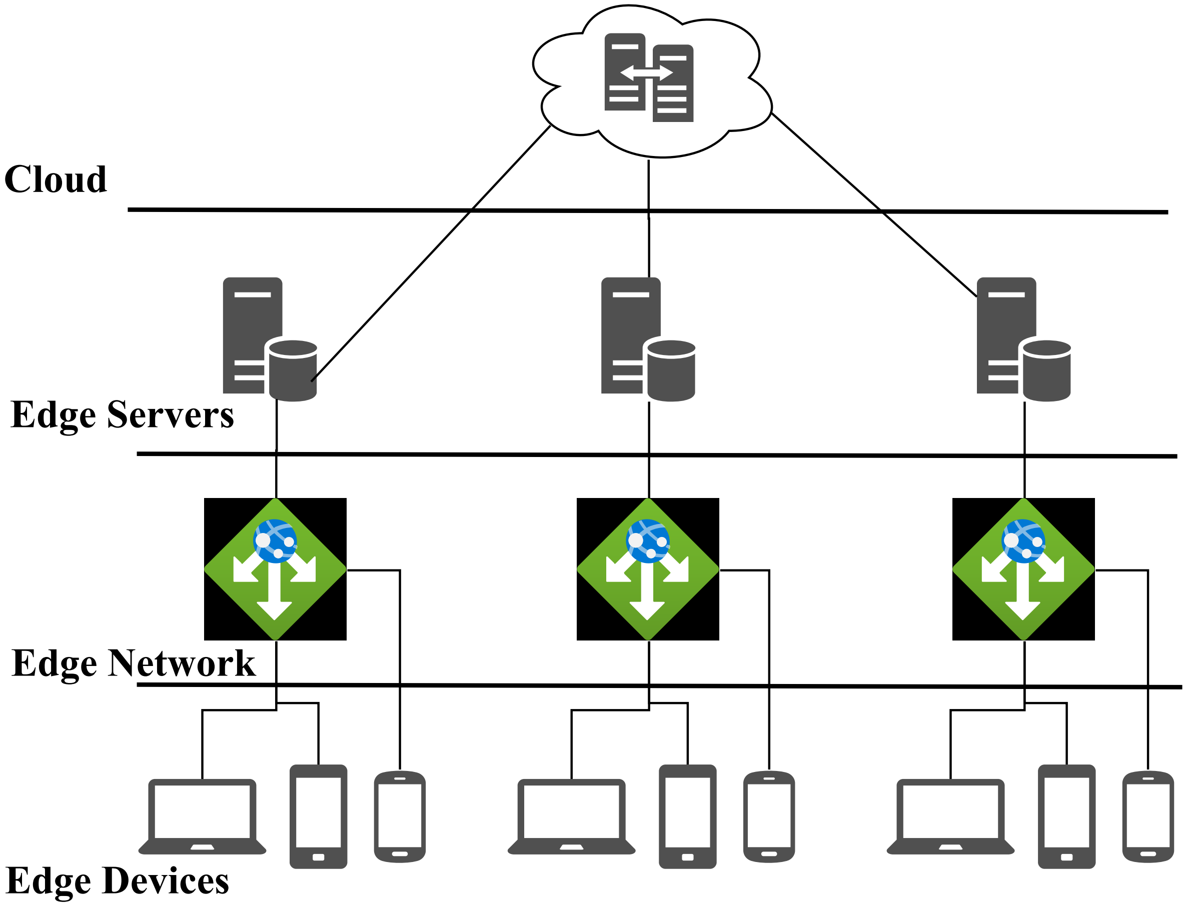

Figure 1: Edge Computing Infrastructure

from IoT devices to edge servers. The proposed scheme uses the Tangle protocol [11, 12] as a cache to reduce the overhead of excessive data processing and task delays among edge servers and IoT devices. The introduction of Tangle as the cache has proven to be more scalable and capable of handling more transactions compared to a traditional cache and a traditional blockchain.

We also propose a secure data transfer mechanism based on reversible integer transformation for the IoT devices to securely transmit device data to an untrusted edge environment. Our method is effective at protecting the privacy of data and preventing information loss.

The main contributions of this paper can be summarized as follows:

1. We propose a mechanism for assisting edge computing for resource-constrained IoT devices by using Tangle as cache.
2. Unlike blockchain alternatives, Tangle uses a DAG data structure that allows transactions to be performed in parallel, making the proposed system scalable.
3. We propose a secure and lightweight data transfer scheme based on reversible integer transformation to transmit data via an unsecured channel to the untrusted edge environment.

The rest of this paper is structured as follows: Section 2 provides a literature overview of previous work. In Section 3, an overview of the preliminaries is provided, followed by the problem definition in

Section 4. The objective and proposed design of our scheme and security requirements are presented in that section. Our proposed methodology is described in Section 5. In Section 6 we provide the security and performance analysis of our schemes under different attack scenarios. Section 7 concludes the paper.

2 Related Work

In this section, we present prior studies that are relevant to this work, categorized under the following three topics: 1) blockchain technology, 2) edge cache services, and 3) privacy preserving computation.

2.1 Blockchain Technology

Blockchain technology has been used widely to store data and share information. Blockchain uses distributed ledger technology (DLT) to store digital records in a decentralized manner. Using cryptographic hashing, these records are ensured to be unalterable [13] Each block in a blockchain contains the hash of the previous block as well as the records in the block itself which guarantees the immutability of the data [14]. Blockchain technology, which has proved beneficial in mitigating mistrust and data loss issues in computation offloading, plays a vital role in effectively establishing trust for edge services to operate in the public ecosystem. In [15] and [16], the authors proposed a method that uses blockchain to preserve data integrity in edge computing.

Smart contracts are used to enhance the functionality of the distributed services alongside blockchain. Using this technology, we can achieve effectiveness and efficiency without involving any third parties. In [17], smart contacts are used to ensure data storage security in vehicular edge networks. In [18], smart contracts are used to enhance system authentication. In [19], smart contracts are used to manage resources in edge computing. However, the focus of these works is to ensure the integrity and solve authentication issues. Few studies have been conducted to enhance the overall efficiency of edge computation. Table 1 provides a comparison of previously proposed approaches in terms of three criteria: (a) whether the approach provides data integrity protection (b) whether the approach is DLT-based (c) whether the approach utilizes the Tangle structure.

2.2 Edge Cache Services

Caching is used at the edge environment to address the reliability and confidentiality issues of data. Edge cache services (ECS) can enhance Quality of Service (QoS) and minimize the cost of transmission of the data. A number of studies [21]-[26] have been conducted on the introduction of caching to the edge environment. The majority of the studies are related to using a cache for edge storage itself. The focus in those studies is to increase the efficiency of the edge servers. However, a small number of studies instead focus on the issues related to integrity, which affects data truthfulness. In [20], rather than using the local storage as a cache, blockchain is used, which provides decentralized storage and proves to be more resilient.

The efficiency issues in edge computing are due to the excessive transmission rate, repeated computation of the data, and job delays among edge servers and IoT devices. The concept of adding a cache at edge servers was introduced to solve edge servers' efficiency issues. In [34], the authors proposed a scheme in which the local storage of an edge server is used as the cache, improving the efficiency of edge computing's overall process. However, [34] does not provide a solution for storage, scalability, and integrity issues. In contrast, a traditional blockchain is used as a cache in [20] to mitigate the storage and integrity issues. However, using a traditional blockchain as the cache causes comparatively high response times and the involvement of miners prevents the scheme from being cost-effective. All of these call for an efficient, scalable, and cost-effective solution to enhance edge computing efficiency.

Table 1: Comparative Analysis Of Related Work with the Proposed Scheme

Ref	**Data Integrity Protection**	**DLT-based**	**Tangle-based**
[15]	✓	✓	x
[17]	✓	✓	x
[20]	✓	✓	x
[21]	✓	x	x
[22]	✓	x	x
[23]	✓	x	x
[24]	✓	x	x
[25]	✓	x	x
[26]	✓	x	x
[27]	x	x	x
[28]	✓	x	x
[29]	x	x	x
[30]	✓	x	x
[31]	x	x	x
[32]	x	✓	x
[33]	x	✓	x
Proposed Scheme	✓	✓	✓

As IoT devices are resource-constrained in terms of computational power and storage, running expensive consensus algorithms for blockchain on IoT devices is not feasible [20]. Blockchain also has scalability issues, and it has a low rate of transactions per second. The approach proposed in this paper uses Tangle [11] as a cache structure. This scheme is proved to provide a better solution than using a traditional blockchain for resource-constrained and lightweight devices.

2.3 Privacy Preserving Computation

In edge computing, user data is typically kept and processed by certain honest-but-suspicious authorized third-party entities, resulting in the leakage of users' personal information. The data can be accessed by an attacker during its transmission as well. Users prefer their data to be secured before it leaves their devices, even when the data collectors cannot get the correct data from users. The authors in [35] utilized encryption technologies to keep consumers' data safe. [36] used the position distortion approach to safeguard users' personal information. However, in actual implementations, the aforementioned investigations, such as k-anonymity [37], [38], and cryptographic protocols, would consume a significant amount of network bandwidth and computing overhead. They are not appropriate for low-power, lightweight edge devices. Furthermore, various data distortion techniques have been frequently employed in the past to change the original data content to preserve data privacy. However, those schemes cause considerable information loss due to distortion in the data.

3 Preliminaries

3.1 W-OTS: The Wintermitz one-time signature scheme

The Wintermitz one-time signature scheme was initially introduced as an extension of Markle's OTS [39]. W-OTS works on the principle of iteratively applying a function to a secret input, with the number of iterations controlled by the message to be signed. The functions that were employed belongs to the following function family.

$$F(n) = \{f_k : \{0,1\}^n \to \{0,1\}^n \mid k \in \{0,1\}^n\} \tag{1}$$

parameterized by the security parameter n and the key $k \in \{0,1\}^n$. The Wintermitz scheme is defined by a parameter w, which is usually a small power of two. The following explanation is designed for the scenario when $w = 2^t$. This can be generalized to any arbitrary w.

We can define:

$$N = \left\lceil \frac{n}{t} \right\rceil + \left\lceil \frac{(\log_2 n - \log_2 t + t)}{t} \right\rceil \tag{2}$$

Key Generation The x_i, for $i = 1....,N$ returns the private key. Then, the public key is calculated by first computing $y_i = f^{2^t-1}(x_i)$ and then computing

$$Y = f(y_1 \parallel y_2 \cdots, \parallel y_n) \tag{3}$$

Signing The message hash is divided into $\lceil \frac{n}{t} \rceil$ segments of length t-bits. The check sign is formed by treating the m values of b_i as integers.

$$C = \sum_{i=1}^{\lceil \frac{n}{t} \rceil} 2^t - b_i \tag{4}$$

Verification Create bi from m using the signature procedure and calculate $v_i = f^{2^t-b_i-1}(s_i)$. The signature is considered to be valid if and only if $Y = f(v_1 \parallel v_2 \cdots \parallel v_n)$.

3.2 Directed Acyclic Graphs

The wide use of Directed Acyclic Graphs or DAGs in successful cryptocurrencies has motivated their use in this work. It should be noted that DAGs are merely a form of data structure, not a blockchain network. The knowledge of how DAG functions can assist readers to more easily comprehend blockchains. Its most successful applications include NXT, IOTA, and IoT Chain. DAGs store the transactions topologically in a graph, unlike real blockchain networks, which store data as a chain of blocks where blocks include the transactions. DAGs have block-less structures, usually referred to as blockchains without the blocks [40].

Storing the full block and verifying the whole chain is not necessarily significant. This ensures that the transactions are verified and added to the network faster compared to PoW and PoS based networks. In DAGs, as long as the information is directed in the same way, nodes can exist in parallel, as opposed to traditional blockchains that follow a strict ordering of transactions.

The network width can be increased by linking a transaction to the previous transaction. The DAG must link an existing and new transaction of the network for a validated transaction. The network chooses a prior transaction to create a link, whenever a transaction is performed. It maintains a certain range for network width, which can support a fast validation process. To control the network width, IOTA presented a new algorithm named Tangle [12].

DAG networks do not depend on special hardware and any mining process, thus have low power consumption. The transactions are validated in an instant. For example, an IoT chain may process over 10,000 transactions per second. It can withstand a 51% assault, making it ideal for IoT and Machine-to-Machine communications.

3.3 Tangle Protocol

Tangle implements the following steps for the signature scheme and constant address generation. Whenever monetary units (IOTAs) are transferred by the user, it transfers the remainder units to a newly generated address. The signing scheme runs based on Winternitz type one-time signature scheme (W-OTS) which originates the later discussed consequences. A transfer is issued by establishing multiple transactions as part of a bundle. It displays the source and receiver address and transactions with the input's transaction signature on it as an input, output, and meta-transaction, respectively. The value of a bundle's transactions is tallied to be kept at 0 to maintain the validity of the bundle.

Three operations are performed for each transaction which includes:

Bundle creation and Signature Meta-transaction stores the signature and creates the bundle when the transaction is being signed using the private key.

Tip selection Two Tips are picked from the Tangle's recent state by using a Random Walk Monte Carlo (RWMC) algorithm. It runs verification process on not only these two transactions but also on the ones that are directly or indirectly referenced. The correctness of the elements of the transactions, for instance signatures, PoW, hashes, etc. and the elimination of conflicts are the primary steps to be taken to verify the transactions.

Proof of Work (POW) For nonce detection and generation of a transaction hash and for the purpose of protection against spam, the required PoW is performed. The Tangle is then updated with a final transaction object that has been transmitted throughout the network. It is then stored in each of the network's Full Nodes.

4 Problem Definition

4.1 System Model

In our proposed system model, we assume an edge computing environment in which three types of entities are involved. These participants include IoT devices (ID_s), Edge Servers (ES_s) and Tangle-based DLT ($IOTA$), as illustrated in Fig.2.

IDs Since IoT devices are resource-constrained devices, they cannot perform complex computations on their own. They need external assistance, namely edge servers, to offload their data for computation. Before sending the data in the edge environment, IoT devices will first distort the original data to preserve privacy.

ESs Edge servers are powerful systems placed at the network's edge where computation is required. They are physically close to the IoT devices or applications that generate the data that the server stores or uses. Edge servers will perform computations as requested by the IoT device and send the results back.

Tangle We use Tangle as a cache to reduce the overhead of excessive data computation and job delays among edge servers and IoT devices. After successful computation, the server will store the results of the computation in Tangle.

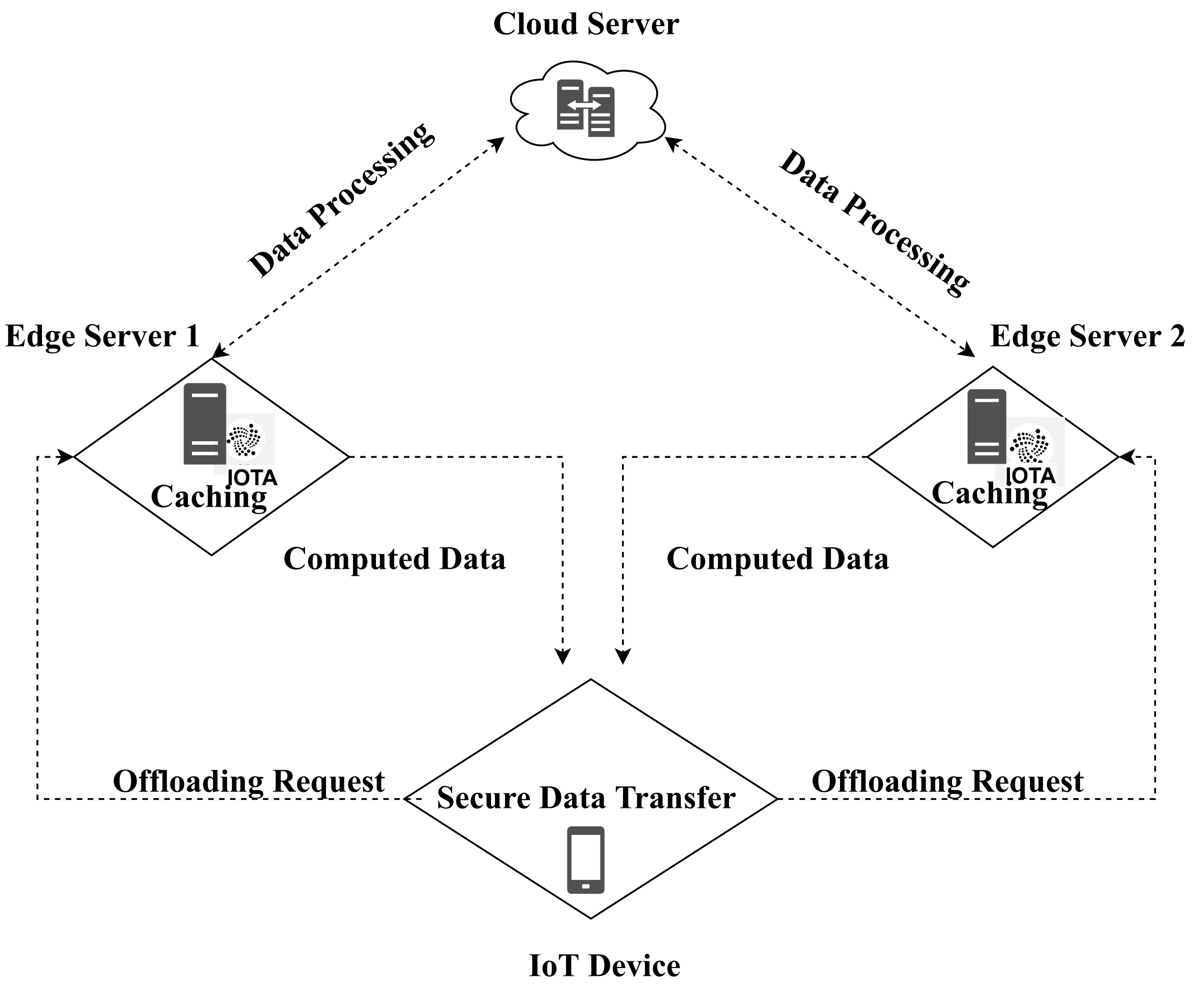

Figure 2: System Model

When the IoT device (ID$_i$) sends a job offloading request to the edge servers, each server will calculate the estimated job execution time based on the cache results and the job itself. Then each server sends back the estimated job execution time to the IoT device (ID$_i$). The device (ID$_i$) then selects the server with the minimum job execution time. The weight value is the sensitivity of the job (J$_i$). Sensitivity is directly proportional to real-time performance.

4.2 Security Requirements

- **Confidentiality:** Confidentiality is a key requirement that guarantees only the data owner and authorized parties have access to the private information. When users' private data is sent and received in the edge, and held or processed in the edge, unauthorized parties should be prohibited from accessing the data.

- **Integrity:** The role of integrity is to guarantee that data is delivered correctly and consistently to the authorised user(s) without any detectable data alteration.

- **Availability:** In edge computing, availability guarantees that all authorized parties may access edge resources at any time and in any location as needed by users. It also implies that the users' data is stored encrypted in the edge and might be handled according to varied operational needs.

- **Authentication:** Authentication guarantees that a user's identity is approved before the user is allowed to interact with the sensitive parts of the system. It is a method of establishing confirmation of a user's identity. Moreover, access control acts as a convergence point for almost every security control and privacy requirement; it dictates who can access the resources (authentication) and what kinds of actions can be performed, such as trying to read (confidentiality) and update/change (integrity).
- **Privacy Requirement:** These security features are employed to ensure that all of the user's outsourced material, including data, personal identity, and location, remains private even from honest but interested parties. Furthermore, data protection methods such as encryption, integrity checking, authentication, and authorization can protect users' privacy in edge computing, either directly or indirectly.

5 Proposed Methodology

In this section, we will briefly describe our proposed schemes; IaC and SDT. IaC is designed for the edge endpoint, it enhances the efficiency of the end-to-end edge computing mechanism. SDT is proposed for the device end to achieve data security and privacy.

5.1 Device end $\Leftrightarrow$ SDT: Secure Data Transfer

In edge computing, user data is typically kept and handled by untrustworthy entities, resulting in the leaking of personal information. What is more, an adversary can access the data during transmission. To address these issues, this paper proposes a secure data transmission technique that protects user privacy by utilizing reversible integer transformation. In this scheme, the device will not send original data D to the edge nodes. Instead, it will send distorted data D' to preserve the privacy of the data. The following is a detailed description of the method.

Data distortion phase:

We will take an original data set D, sensitive attributes $S = (p_k)$ where $k = 1, 2, 3, \cdots$, an integer seed, a group size gp, weights w_i where $(i \in [0, gp-1])$ and watermark bits wm.

- **Step 1:** Let $n = \left\lfloor \frac{|D|}{gp} \right\rfloor - 1$ and $l = 1$.
- **Step 2:** For each p_k:
  1. Let $\left\langle p_{k,x}, p_{k,x+1}, p_{k,x+2}, \cdots, p_{k,x+(gp-1)} \right\rangle$ be a group of gp adjacent data values $(x = (1, 1+(1 \times gp), 1+(2 \times gp), \cdots, 1+(1 \times n)))$.
  2. Use Eq. (5) and Eq. (6), $\left\langle p_{k,x}, p_{k,x+1}, p_{k,x+2}, \cdots, p_{k,x+(gp-1)} \right\rangle$ and get $\left\langle \tilde{p}_{k,x}, \tilde{p}_{k,x+1}, \tilde{p}_{k,x+2}, \cdots, \tilde{p}_{k,x+(gp-1)} \right\rangle$

$$p'_{k,x} = \left\lfloor \frac{w_0 p_{k,x} + w_1 p_{k,x+1} + \cdots + w_{gp-1} p_{k,x} + (gp - i)}{w_0 + w_1 + w_2 + \cdots + w_{gp-1}} \right\rfloor \tag{5}$$

$$p'_{k,x+1} = p_{k,x+1} - p_{k,x}$$

$$p'_{k,x+2} = p_{k,x+2} - p_{k,x}$$

$$\cdots,$$

$$p'_{k,x+(gp-1)} = p_{k,x+(gp-1)} - p_{k,x}$$

$$\tilde{p}_{k,x} = p'_{k,x} \tag{6}$$

$$\tilde{p}_{k,x+1} = 2 \times p'_{k,x+1}$$

$$\tilde{p}_{k,x+2} = 2 \times p'_{k,x+2}$$

$$\cdots,$$

$$\tilde{p}_{k,x+(gp-1)} = 2 \times p'{}_{k,x+(gp-1)}$$

3. If, $l \leq |wm|$
   then for $\tilde{p}_{k,x+1}, \tilde{p}_{k,x+2}, \ldots, \tilde{p}_{k,x+(gp-1)}$ in $\left\langle \tilde{p}_{k,x}, \tilde{p}_{k,x+1}, \tilde{p}_{k,x+2}, \ldots, \tilde{p}_{k,x+(gp-1)} \right\rangle$ we attach the l^{th} bit, the $(l+1)^{th}$ bit, the $(l+(gp–2))^{th}$ bit of the watermark wm respectively. $l = l + (gp-1)$
4. Use Eq. (7) to generate the corresponding distorted group.

$$p''_{k,x} = \tilde{p}_{k,x} - \left\lfloor \frac{w_1 \tilde{p}_{k,x+1} + \cdots + w_{gp-1} \tilde{p}_{k,x} + (gp-i)}{w_0 + w_1 + w_2 + \cdots + w_{gp-1}} \right\rfloor \tag{7}$$

$$p''_{k,x+1} = \tilde{p}_{k,x+1} + q''_{k,x}$$

$$p''_{k,x+2} = \tilde{p}_{k,x+2} + q''_{k,x}$$

$$\cdots,$$

$$p''_{k,x+(gp-1)} = \tilde{p}_{k,x+(gp-1)} + p''_{k,x}$$

- **Step 3:** To create random values of $|D|$, use a random `Seed` function. To produce the distorted data D'', the distorted data D' is sorted in ascending or descending order depending on the values of these random variables.

5.2 Edge end ⇔ IaC: Tangle as Cache

Our scheme proposes a cache-aware computation offloading methodology based on Tangle.

This scheme will be implemented on the edge devices (i.e. servers), which provide secure and efficient computation offloading. IoT devices (ID_i) perform a relatively low amount of computation locally according to their capacity but use the edge and cloud servers for complex computations. This brings some security and performance-related issues to the edge server's domain. Our focus is to enhance the computation's security and efficiency on the edge servers. Our proposed scheme achieves integrity and cost-effectiveness with Tangle's help and using Tangle as a cache we achieve efficiency in computation offloading.

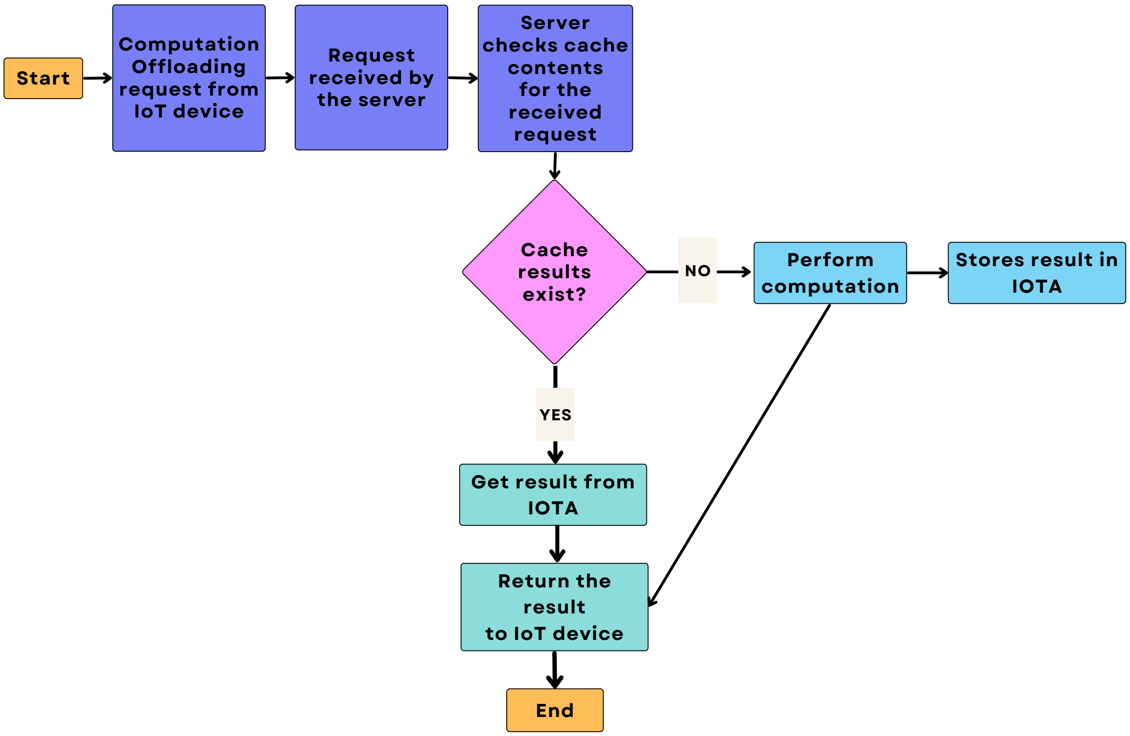
Start
Computation Offloading request from IoT device
Request received by the server
Server checks cache contents for the received request
Cache results exist?
NO
Perform computation
Stores result in IOTA
YES
Get result from IOTA
Return the result to IoT device
End

Figure 3: Flow Diagram of our Proposed Scheme IaC

The overall mechanism of IaC is illustrated in Fig.3. The following are the phases involved in our proposed cache-based computation offloading scheme:

- Offloading Request Phase: When an IoT device needs to perform any computation, it will send the computation request to the edge servers. However, before transmitting the original data to the edge environment, the device needs to convert the original data D to D' by implementing the SDT scheme mentioned above.
- Server Selection Phase: When an IoT device (ID_i) broadcasts the computation request (J_i) to all the edge servers, the system needs to check some conditions for an optimal server selection.

Let $ES = 0, 1, 2, \ldots, N$ be the set of edge servers that can perform the computation. They receive the computation request r from the IoT device, as a set of transaction requests $J = 0, 1, 2, \ldots, j$. Then the IoT device sends the request $j \in J$ to ES_i, such that: $i \in ES$. After receiving the request r, the selection of ES_i is decided on the basis of the weighted cost of each ES_i. Here, it is important to mention that the weighted cost ΔV depends on two factors; load (L_t), and record of cache (I_t).

When the IoT device (ID_i) needs to perform computation offloading, it sends a job offloading request (J_i) to all the servers. Then, all the edge servers will check their cache to see if they have fulfilled this request (J_i) before and whether the computed data Dc request exists in the cache or not. If any server has already fulfilled the same request (J_i) before, the computed data (Dc_i) is in the cache. Then, there is no need to compute on the data (D_i') again and the server will resend the computed data from the cache to the IoT device (ID_i).

The server selection is based on the two criteria mentioned below:

1. Cache Content: The following are the steps involved in the selection of the server based on the cached data:
   (a) An IoT device broadcasts the offloading request (J_i) to the edge servers.
   (b) All servers check for requested data (d_i) in their cache.
   (c) The server with the cached content is selected.
   Cache: $I_t = I_t^i, \forall i \in ES$ so that each server ES_i first verifies the computation request r from the cache. If the record for the request J_i exists then the flag is set to 0 as no pre-computation is required for this request. Otherwise, the flag is set to 1 which represents that this request does not exist in cache I_t^i, such that, $I_t^i \in [0, 1]$. We can say that, $\forall ES_t^i, I_t^i = J_t \Leftrightarrow 0$
   $$\Delta I_t^{ES_i} = [0, 1]$$
   (d) Then that server returns the computed data to the IoT device (ID_i).

2. Closest Server/Server with less load: The following are the steps involved in the selection of the server based on the nearest server criterion:
   (a) An IoT device broadcasts the offloading request (J_i) to the edge servers.
   (b) All servers check for the requested data (D_i') in their cache.
   (c) When no server with the cached content exists, then the server with the smallest response time (i.e. nearest server/server with the least load) is selected to respond to the request.
   Load: L_t= $L_t^{(i)}$, $\forall i \in ES$ represents that the load at E_i when it receives the request J at time t. If ES is already loaded with the computation request and cannot perform the computation until completion of already queued requests, this can be represented as L and if the server is free to perform computation on received J then it is represented as 0. So, $L_t^i \in [0, 1]$. $\forall ES, \exists l_t \Leftrightarrow \leq 0$. Hence the load on servers can be determined using the following equation:
   $$\gamma_t^i = \sum_{i \in j}^{|j|} (t_1^{j_i} + t_2^{j_{i+1}} + j_3^{r_{i+2}}, \ldots j_n^{|r|}) \tag{8}$$
   $$\Delta L_t^{ES_i} = \sum_{i \in J} 1_{\gamma_t^i} \forall E \in 0, 1, 2, \ldots, N \tag{9}$$
   Here in Equation 8, the total time denotes the time required for all the current requests that the server is working on.

(d) Then that server performs the computational and returns the computed data to the IoT device (ID_i).

Each server will calculate the weight ΔV based on the cache record ΔI and load ΔL

$$\Delta V^{ES_i} = \Delta I_t^{ES_i} + \Delta L_t^{ES_i} \tag{10}$$

The server with the minimum value of ΔV^{ES_i} will be selected.

- Caching Phase: After successful computation on the data, the computed data Dc will be stored in Tangle. Next time, when a user requests the same data, there will be no need to compute on the data again, the server will simply retrieve that data from Tangle and send it to the IoT device.

6 Security and Performance Analysis

In this section, we provide security and performance analyses of our proposed schemes.

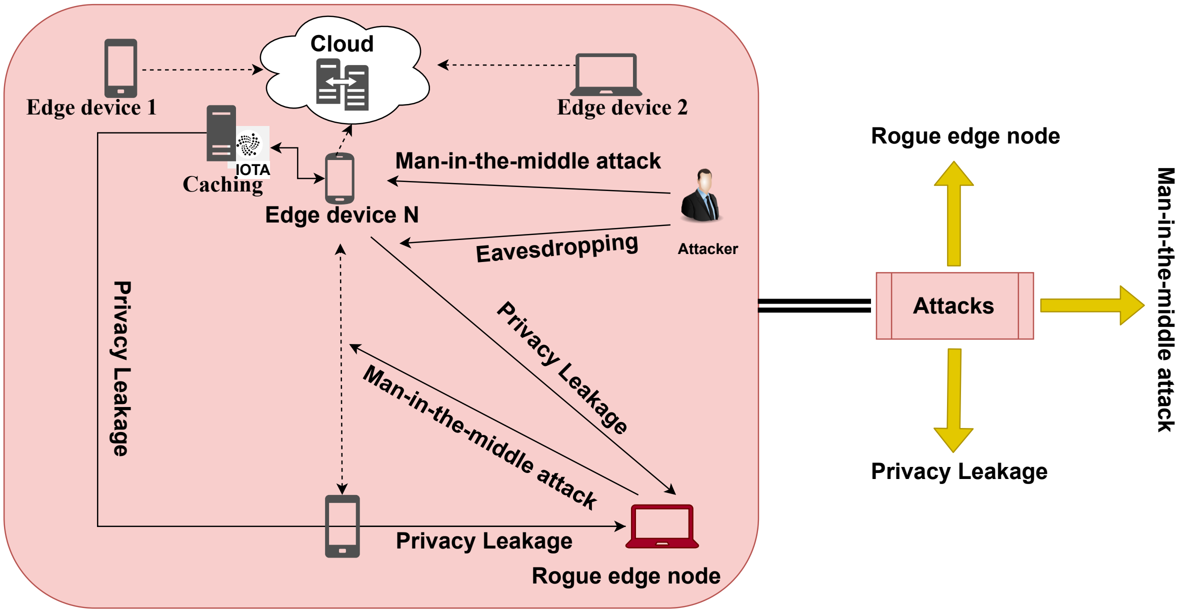

Figure 4: Attack Model in Edge Computing

6.1 Security Analysis

The security criteria that our approach can meet are discussed in this section. This is derived from Tangle's and the SDT scheme's security. We have evaluated our security mechanism against various types of adversarial attacks illustrated in Fig.4.

- Data Leakage: In our proposed scheme (SDT), we first distort the original data D to D' so that the distorted data is transferred to the untrusted edge environment. Thus, the privacy of the original data and the data sender is preserved. If an attacker captures the data during the transfer, distorted data will be of no use to the attacker. The following equation is used to produce the associated distorted group.

$$p''_{k,x} = \tilde{p}_{k,x} - \left\lfloor \frac{w_1 \tilde{p}_{k,x+1} + \cdots + w_{gp-1} \tilde{p}_{k,x} + (gp-i)}{w_0 + w_1 + w_2 + \cdots + w_{gp-1}} \right\rfloor$$

- Rogue Edge Node: At the edge end, there might be a malicious edge node which can be a threat to the privacy and confidentiality of the data. The privacy of the data is preserved by our proposed scheme SDT and the confidentiality of the data is ensured by implementing the private MAM layer. Encoded streams that are not intended for public access can be played in the private mode. The Merkle root hash is utilised as the address in private mode. Because the attacker cannot deduce the root from the hash, this prevents an adversary from decoding the communication if they come across it.

- Quantum Computing Threats: Tangle utilizes hash-based signatures rather than elliptic curve cryptography (ECC). This differs from Bitcoin in the sense that it makes the protocol resilient to Quantum Cryptography while also making signing and validating transactions easier and faster. Because of "Winternitz signatures", Tangle is quantum-resilient.

  - Key Generation:

    $$Y = f(y_1 \parallel y_2 \cdots, \parallel y_n)$$

  - Signing:

    $$C = \sum_{i=1}^{\lceil \frac{n}{t} \rceil} 2^t - b_i$$

6.2 Performance Analysis

6.2.1 STD Performance Analysis

In this section, we analyze the performance and accuracy of our proposed scheme. We have compared actual data and distorted data and analysed the deviation. We have also compared the distorted data with the expected data. Simulation results as illustrated in Fig. 5 show that SDT is more effective at preserving the privacy of original data and preventing data loss.Also we have analysed the behaviour of our proposed scheme by changing the weights. We may improve the flexibility of privacy-preserving computation by employing a customizable weighting factor to quantify the level of distortion of the original data. Simulation results in Fig. 6 show that by increasing the value of the weights, the privacy of the data increases while the utility of the data decreases due to increased deviation.

6.2.2 IaC Performance Analysis

The performance of our proposed scheme was analyzed by setting up a blockchain server, Tangle server and a cache server.

- Experimental Setup: Tools used to implement the scheme and determining its feasibility and functionality are "Node.js", "Ganache CLI" and "MongoDB". We have used a third-party API "JSON-Placeholder" for the data.

- Efficiency: We first set up servers on the Google Cloud Platform for the experiments. Then, we wrote an API to send and handle data requests. For demonstration purposes, we have set up 5 servers. Two of them use Tangle as the cache and the other two use a traditional blockchain (the

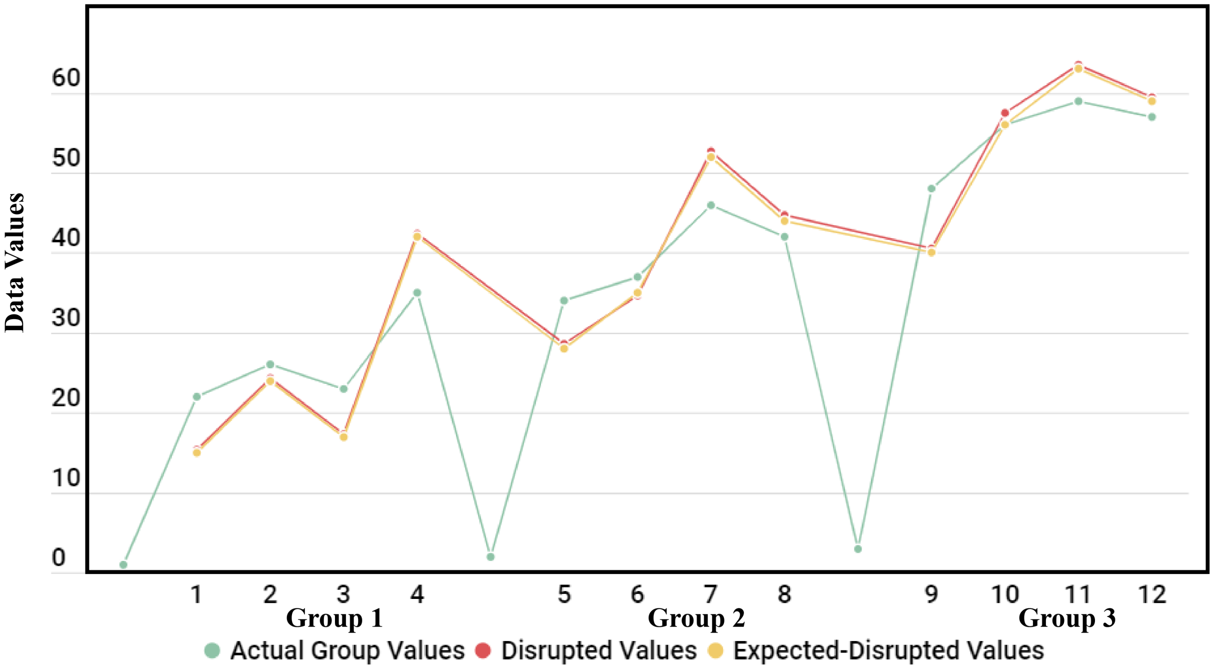

Figure 5: The comparison results of SDT data distortion with actual data under different groups

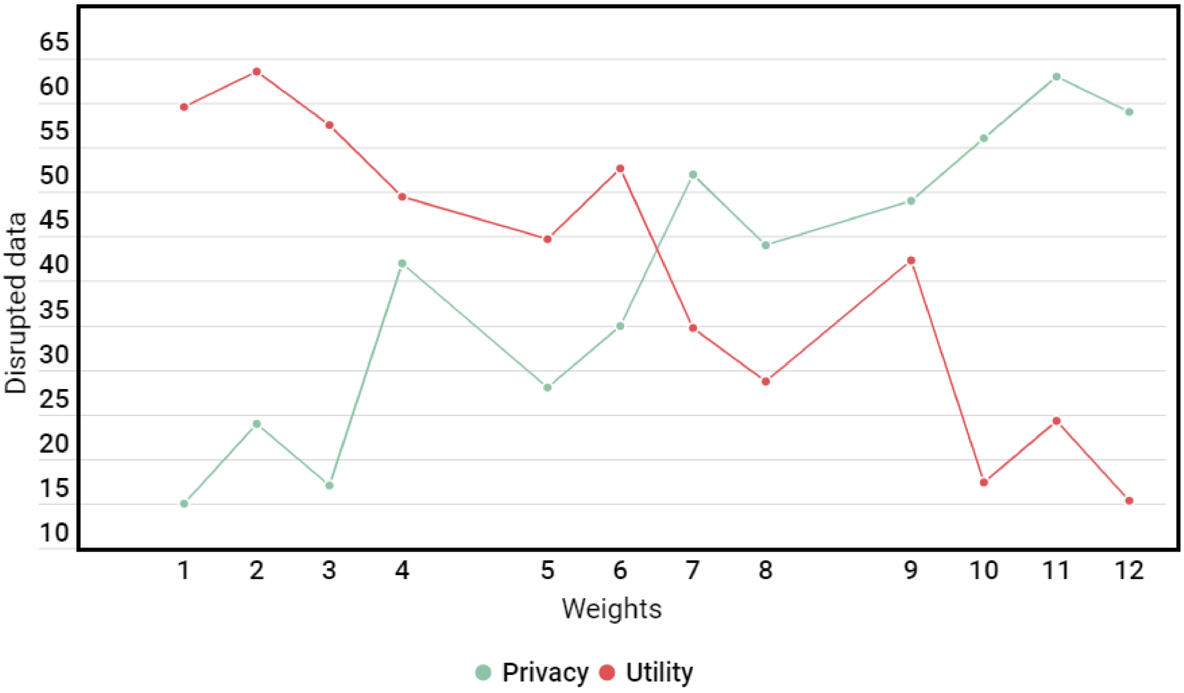

Figure 6: Utility vs privacy graph when weights are applied to the data

blockchain structure underlying Bitcoin) as the cache and the remaining server uses local storage as the cache.

Experiments proved that the response times for the servers, which use the Tangle protocol, are much less than those of the servers using the blockchain protocol [20].

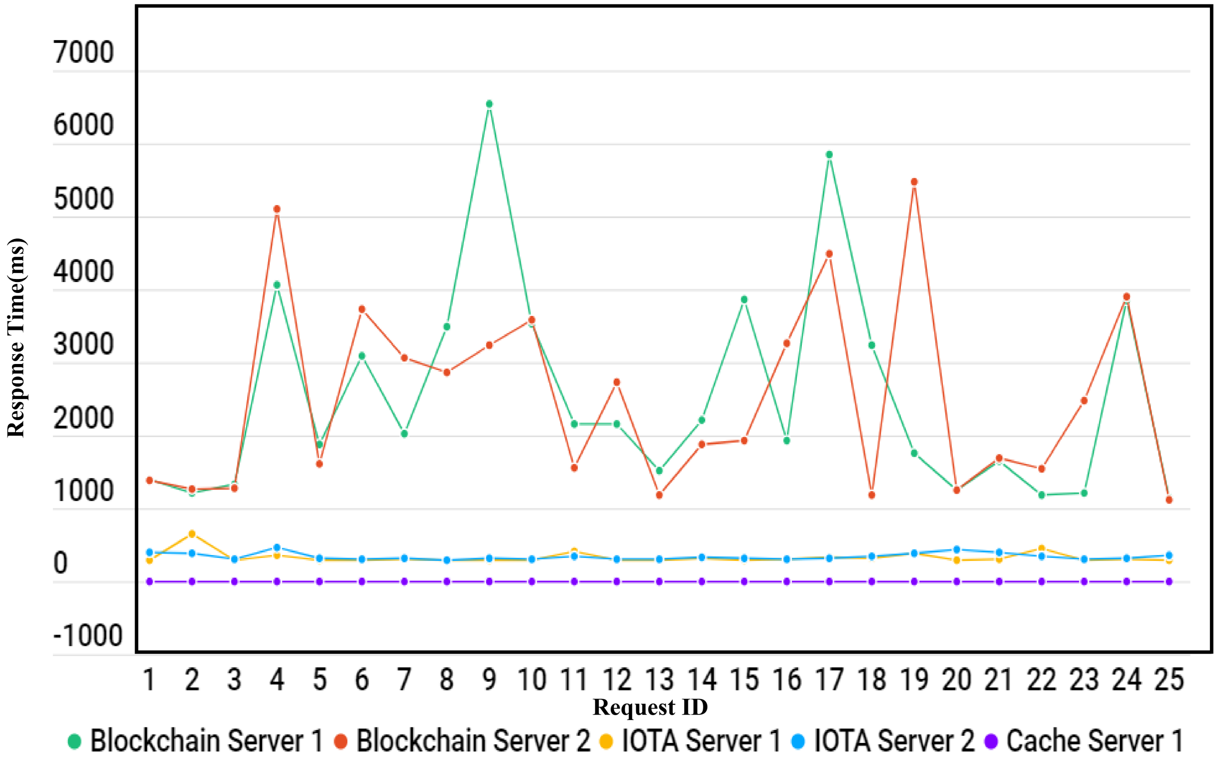

Figure 7: Performance comparison of our work with [20] and [34]

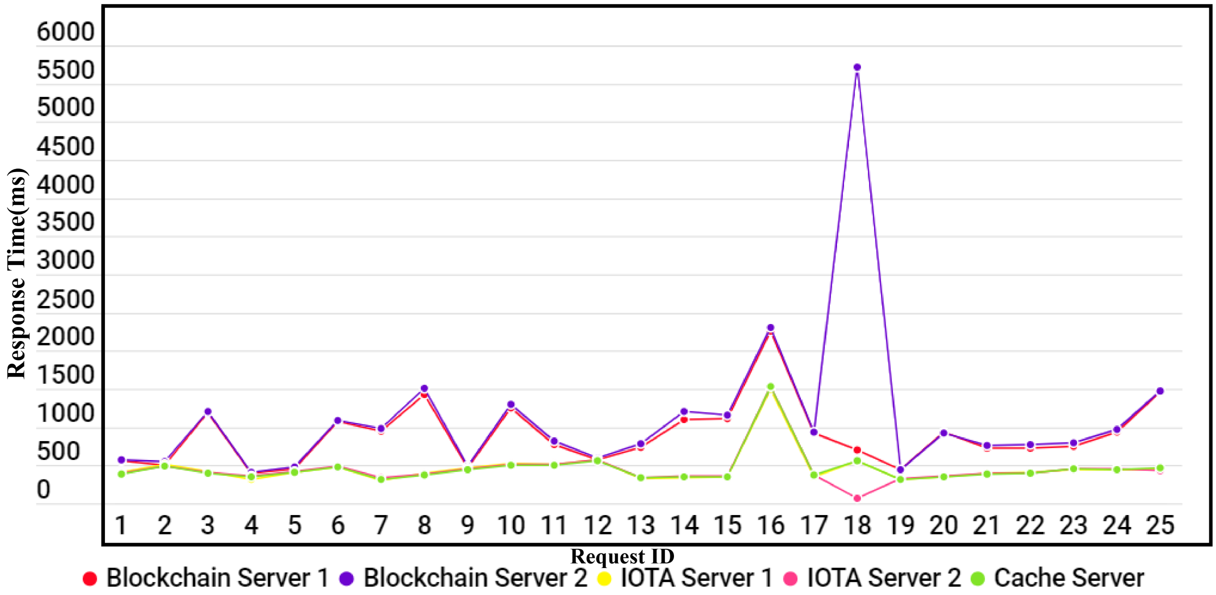

Figure 8: Extended performance comparison of our work with [20] and [34]

We have also compared our proposal with the method presented in [34], where part of the local storage of the edge servers is used as the cache. The response time of our scheme is relatively high compared to [34]. However, the said study presents a chance of tampering with data. On the other hand, our scheme provides integrity assurance because of the Tangle structure. The performance comparison of our work with [20] and [34] is provided in Fig. 7.

- Scalability: Scalability is a fundamental problem a distributed cache has to address. A scalable cache is one that can maintain the desired performance even when the transaction load on it increases. In order to compare the scalabilities of the proposed approach and that of blockchain, we performed experiments where batch data requests were submitted to both servers. As Fig. 9 demonstrates, our scheme is more scalable compared to the utilized blockchain approach.

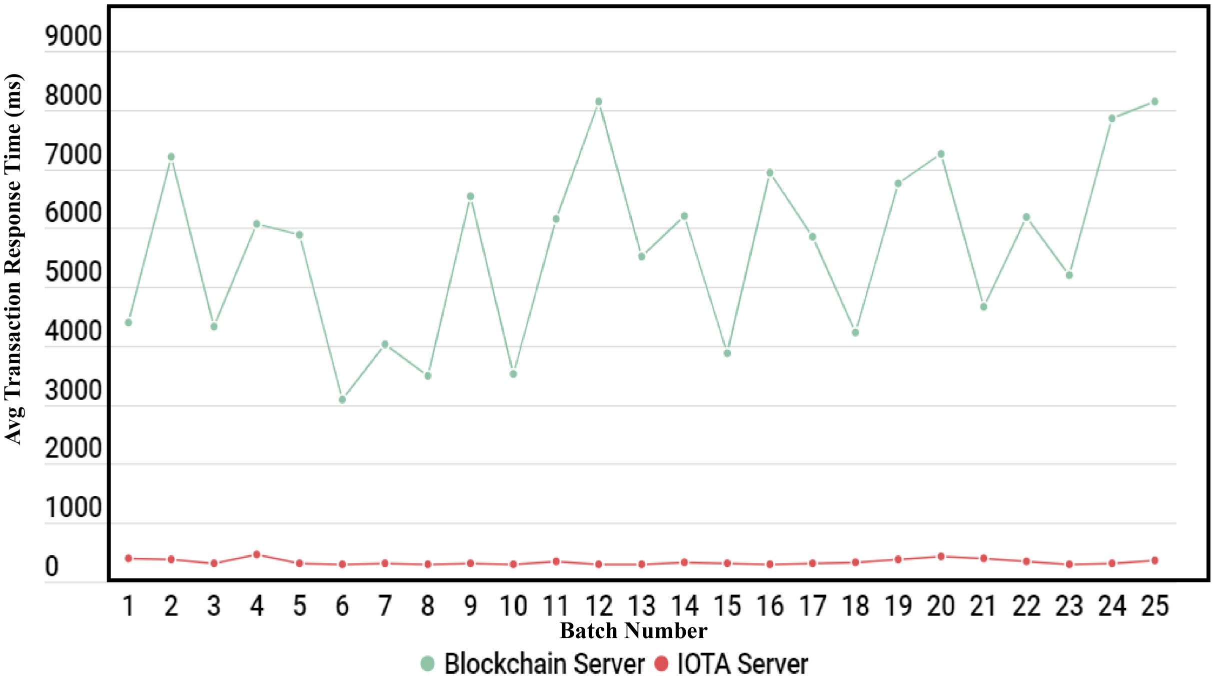

Figure 9: Comparison of scalability of Tangle with Blockchain

7 Conclusion

In this paper, we have proposed a scheme which aids the process of computation offloading in edge computing to enhance efficiency in IoT environments. We have reduced the transmission delay and excessive computation load, and enhanced efficiency in edge computing by introducing Tangle as the cache structure on the edge servers. Our proposed scheme is scalable since Tangle uses the DAG data structure, which supports the addition of transactions in parallel. The simulation results show that our scheme can reduce the response time dramatically compared to other schemes and our proposed scheme is more scalable than a PoW-based blockchain approach.

We have also proposed a secure and lightweight data transfer scheme based on reversible integer transformation to transmit the data via an unsecured channel to an untrusted edge environment. SDT is effective at preserving the privacy and integrity of the original data and preventing data loss.

Our future work will involve extending the experimental study to include a variety of blockchain platforms and larger loads for testing the scalability.

References

[1] Ericsson. About ericsson - corporate information., December 2022. `https://www.ericsson.com/en/about-us` [Online; Accessed on December 8, 2022].

[2] K. Zhang, S. Leng, X. Peng, L. Pan, S. Maharjan, and Y. Zhang. Artificial intelligence inspired transmission scheduling in cognitive vehicular communications and networks. *IEEE Internet of Things Journal*, 6(2):1987–1997, April 2019.

[3] J. Xu, S. Wang, A. Zhou, and F. Yang. Edgence: A blockchain-enabled edge-computing platform for intelligent iot-based dapps. *China Communications*, 17(2):78–87, April 2020.

[4] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas. Ddos in the iot: Mirai and other botnets. *Computer*, 50(7):80–84, July 2017.

[5] B. Al-Duwairi, W. Al-Kahla, M. Al-Refai, Y. Abedalqader, A. Rawash, and R. Fahmawi. Siem-based detection and mitigation of iot-botnet ddos attacks. *International Journal of Electrical and Computer Engineering*, 10(2):2182–2191, April 2020.

[6] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder. *Bitcoin and Cryptocurrency Technologies*. Princeton University Press, 2016.

[7] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2008. `https://bitcoin.org/bitcoin.pdf` [Online; Accessed on December 15, 2022].

[8] N. Szabo. Smart contracts : Building blocks for digital markets, 1996. `https://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart_contracts_2.html`[Online; Accessed on December 16, 2022].

[9] K. Christidis and M. Devetsikiotis. Blockchains and smart contracts for the internet of things. *IEEE Access*, 4:2292–2303, 2016.

[10] H. Wei, H. Luo, and Y. Sun. Mobility-aware service caching in mobile edge computing for internet of things. *Sensors*, 20(3):1–8, 2020.

[11] M. Conti, G. Kumar, P. Nerurkar, R. Saha, and L. Vigneri. A survey on security challenges and solutions in the iota. *Journal of Network and Computer Applications*, 203:1–22, July 2022.

[12] W.F. Silvano and R. Marcelino. Iota tangle: A cryptocurrency to communicate internet-of-things data. *Future Generation Computer Systems*, 112:307–319, November 2020.

[13] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang. An overview of blockchain technology: Architecture, consensus, and future trends. In *Proc. of the 2017 IEEE International Congress on Big Data (ICBD'17), Honolulu, HI, USA*, pages 557–564. IEEE, June 2017.

[14] T. Ahram, A. Sargolzaei, S. Sargolzaei, J. Daniels, and B. Amaba. Blockchain technology innovations. In *Proc. of the 2017 IEEE Technology & Engineering Management Conference (TEMSCON'17), Santa Clara, CA, USA*, pages 137–141. IEEE, June 2017.

[15] X. Xu, Y. Chen, X. Zhang, Q. Liu, X. Liu, and L. Qi. A blockchain-based computation offloading method for edge computing in 5g networks. *Software: Practice and Experience*, 51(10):2015–2032, October 2021.

[16] R. Casado-Vara, F. de la Prieta, J. Prieto, and J. M. Corchado. Blockchain framework for iot data quality via edge computing. In *Proc. of the 1st Workshop on Blockchain-Enabled Networked Sensor Systems (BlockSys'18), Shenzhen, China*, pages 19–24. ACM, November 2018.

[17] J. Kang, X. Huang, M. Wu, S. Maharjan, S. Xie, and Y. Zhang. Blockchain for secure and efficient data sharing in vehicular edge computing and networks. *IEEE Internet of Things Journal*, 6(3):4660–4670, October 2019.

[18] S. Guo, X. Hu, S. Guo, X. Qiu, and F. Qi. Blockchain meets edge computing: A distributed and trusted authentication system. *IEEE Transactions on Industrial Informatics*, 16(3):1972–1983, March 2020.

[19] K.L. Wright, M. Martinez, U. Chadha, and B. Krishnamachari. Smartedge: A smart contract for edge computing. In *Proc. of the 2018 IEEE International Conference on Internet of Things (iThings'18) and IEEE Green Computing and Communications (GreenCom'18) and IEEE Cyber, Physical and Social Computing (CPSCom'18) and IEEE Smart Data (SmartData'18), Halifax, NS, Canada*, pages 1685–1690. IEEE, July-August 2018.

[20] J. Liu, S. Guo, Y. Shi, L. Feng, and C. Wang. Decentralized caching framework toward edge network based

on blockchain. *IEEE Internet of Things Journal*, 7(9):9158–9174, September 2020.

[21] S. Zhang, P. He, K. Suto, P. Yang, L. Zhao, Xuemin, and X. Shen. Cooperative edge caching in user-centric clustered mobile networks. *IEEE Transactions on Mobile Computing*, 17(8):1791–1805, August 2018.

[22] C. Li, L. Toni, J. Zou, H. Xiong, and P. Frossard. Qoe-driven mobile edge caching placement for adaptive video streaming. *IEEE Transactions on Multimedia*, 20(4):965–984, September 2017.

[23] P. Yang, N. Zhang, S. Zhang, L. Yu, J. Zhang, and X. Shen. Content popularity prediction towards location-aware mobile edge caching. *IEEE Transactions on Multimedia*, 21(4):915–929, April 2019.

[24] A. Mahmood, C. Casetti, C.F. Chiasserini, P. Giaccone, and J. Harri. Mobility-aware edge caching for connected cars. In *Proc of the 12th Annual Conference on Wireless On-demand Network Systems and Services (WONS'16), Cortina d'Ampezzo, Italy*, pages 1–8. IEEE, January 2016.

[25] G. Ma, Z. Wang, M. Zhang, J. Ye, M. Chen, and W. Zhu. Understanding performance of edge content caching for mobile video streaming. *IEEE Journal on Selected Areas in Communications*, 35(5):1076–1089, May 2017.

[26] F. Gabry, V. Bioglio, and I. Land. On energy-efficient edge caching in heterogeneous networks. *IEEE Journal on Selected Areas in Communications*, 34(12):3288–3298, September 2016.

[27] X. Cao, J. Zhang, and H. V. Poor. An optimal auction mechanism for mobile edge caching. In *Proc. of the 38th IEEE International Conference on Distributed Computing Systems (ICDCS'18), Vienna, Austria*, pages 388–399. IEEE, July 2018.

[28] H. Pang, L. Gao, and L. Sun. Joint optimization of data sponsoring and edge caching for mobile video delivery. In *Proc. of the 2016 IEEE Global Communications Conference (GLOBECOM'16), Washington, DC, USA*, pages 1–7. IEEE, November 2016.

[29] J. Du, L. Zhao, J. Feng, X. Chu, and F.R. Yu. Economical revenue maximization in cache enhanced mobile edge computing. In *Proc. of the 2018 IEEE International Conference on Communications (ICC'18), Kansas City, MO, USA*, pages 1–6. IEEE, May 2018.

[30] J. Krolikowski, A. Giovanidis, and M. Di-Renzo. A decomposition framework for optimal edge-cache leasing. *IEEE Journal on Selected Areas in Communications*, 36(6):1345–1359, June 2018.

[31] F. De-Pellegrini, A. Massaro, L. Goratti, and R. El-Azouzi. A pricing scheme for content caching in 5g mobile edge clouds. In *Proc. of the 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM'16), Fez, Morocco*, pages 193–198. IEEE, October 2016.

[32] M. Liu, F.R. Yu, Y. Teng, V.C.M. Leung, and M. Song. Computation offloading and content caching in wireless blockchain networks with mobile edge computing. *IEEE Transactions on Vehicular Technology*, 67(11):11008–11021, November 2018.

[33] Y. Liu, F.R. Yu, X. Li, H. Ji, and V.C.M. Leung. Resource allocation for video transcoding and delivery based on mobile edge computing and blockchain. In *Proc. of the 2018 IEEE Global Communications Conference (GLOBECOM'18), Abu Dhabi, United Arab Emirates*, pages 1–6. IEEE, December 2018.

[34] H. Wei, H. Luo, Y. Sun, and M.S. Obaidat. Cache-aware computation offloading in iot systems. *IEEE Systems Journal*, 14(1):61–72, March 2019.

[35] A. Rohilla, M. Khurana, and L. Singh. Location privacy using homomorphic encryption over cloud. *International Journal of Computer Network and Information Security*, 9(8):32–40, August 2017.

[36] Z. Cai, Z. He, X. Guan, and Y. Li. Collective data-sanitization for preventing sensitive information inference attacks in social networks. *IEEE Transactions on Dependable and Secure Computing*, 15(4):577–590, July-August 2018.

[37] J. Wang, Z. Cai, and J. Yu. Achieving personalized k-anonymity-based content privacy for autonomous vehicles in cps. *IEEE Transactions on Industrial Informatics*, 16(6):4242–4251, June 2020.

[38] Y. Wang, Y. Li, Z. Chi, and X. Tong. The truthful evolution and incentive for large-scale mobile crowd sensing networks. *IEEE Access*, 6:51187–51199, September 2018.

[39] R. Merkle. A certified digital signature. In *Proc. of the 9th Annual International Cryptology Conference (CRYPTO'89), Santa Barbara, California, USA*, volume 435 of *Lecture Notes in Computer Science*, pages 218–238. Springer-Verlag, August 1989.

[40] P. Schueffel. Alternative distributed ledger technologies blockchain vs. tangle vs. hashgraph - a high-level

overview and comparison. *SSRN Electronic Journal*, 4:1–8, March 2018.

Author Biography

Aqsa Ayub received the BS degree in computer science from Quaid-i-Azam University Islamabad, Pakistan, in 2017. She received the MS degree in information security from COMSATS University Islamabad, Pakistan, in 2021. She is currently a Lecturer with the Department of Computer Sciences at COMSATS University Islamabad, Islamabad, Pakistan. Her research interests include edge computing, IOTA, blockchain, and cryptography.

Muhammad Rizwan received a Bachelor's degree from Government College University Faisalabad in 2007 and a Master's degree from Quaid I Azam University, Islamabad, Pakistan, in 2010. From 2020 to 2022, he worked as a researcher at the Southern University of Science and Technology (SUSTech) in Shenzhen, China. He is currently working as a Research Scholar in the Computer Science and Technology department at the University of Science and Technology of China (USTC). His research interests include cloud computing, IoT security, distributed systems, 5G networks, Game Theory, and blockchain

Shivanshu Shrivastava (Member, IEEE) received the B.Eng. degree in Electronics and Telecommunications from Shri Shankaracharya College of Engineering and Technology, CSVTU Bhilai, India, in 2010, and the Ph.D. degree in Communication Engineering from the Indian Institute of Technology (IIT) Guwahati, India, in 2017. From 2017 to 2018, he was with the Department of Electrical Engineering, IIT Kanpur, as the Project Investigator of the Science and Engineering Research Board Project titled "Designing Energy Efficient Hybrid RF/VLC-Based CRANs for 5G. " From 2019 to 2021, he was with the College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China, as a Postdoctoral Researcher. Since 2021, he has been working as an Assistant Professor with the Department of Electronics Engineering, Rajiv Gandhi Institute of Petroleum Technology (An Institute of National Importance), Amethi, India. His research interests include Deep Learning Applications in Communication, specifically on throughput maximization in visible light communications for 5G, cognitive radios, and general wireless communication systems.

Adeel Anjum received the Ph.D. degree in computer sciences from the University of Nantes, Nantes, France. He is currently an Associate Professor with the Department of Information Technology, Quaid-i-Azam University, Islamabad, Pakistan, and a Research Assistant Professor with the Southern University of Sciences and Technology (SUSTECH), Shenzhen, China. His research interests include access control systems, model-driven architecture, and work flow management systems.

Pelin Angin (Member, IEEE) received the B.S. degree in computer engineering at Bilkent University, in 2007, and the Ph.D. degree in computer science from Purdue University, USA, in 2013. From 2014 to 2016, she worked as a Visiting Assistant Professor and a Postdoctoral Researcher at Purdue University. She is currently an Assistant Professor in computer engineering at Middle East Technical University. Her research interests include the fields of cloud computing, the IoT security, distributed systems, 5G networks, data mining, and blockchain. She is among the founding members of the Systems Security Research Laboratory and an affiliate of the Wireless Systems, Networks and Cybersecurity Laboratory, METU.

Yigit Sever received the B.S. degree in computer engineering at TED University, Turkey, in 2016, and the M.S. degree in computer engineering from Hacettepe University, Turkey, in 2019. He is currently a Ph.D. candidate in computer engineering at Middle East Technical University (METU), Turkey, where he is also working as a research assistant since 2020. His research interests include cloud security with a strong focus on container security, user and internet privacy and distributed systems. He is a member of Wireless Systems, Networks and Cybersecurity Laboratory, METU.