Wireless Communications and Mobile Computing

Heterogeneous Systems and Algorithms for Next Generation Information Centric Networks, Internet of Things, and 5G Systems

Lead Guest Editor: Muhammad Shafiq
Guest Editors: Zhihong Tian and Habib Ullah Khan

Heterogeneous Systems and Algorithms for Next Generation Information Centric Networks, Internet of Things, and 5G Systems

Wireless Communications and Mobile Computing

Heterogeneous Systems and Algorithms for Next Generation Information Centric Networks, Internet of Things, and 5G Systems

Lead Guest Editor: Muhammad Shafiq
Guest Editors: Zhihong Tian and Habib Ullah Khan

Chief Editor

Associate Editors

Academic Editors

Contents

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5648765, 13 pages
https://doi.org/10.1155/2021/5648765

Research Article

Information-Centric Networking Cache Placement Method Based on Cache Node Status and Location

Dapeng Man,[1] **Yao Wang**,[1] **Hanbo Wang**,[1] **Jiafei Guo**,[1] **Jiguang Lv**,[1] **Shichang Xuan**,[1] **and Wu Yang**[1,2]

[1] *Information Security Research Center, Harbin Engineering University, Harbin 150001, China*
[2] *Peng Cheng Laboratory, 518055, China*

Correspondence should be addressed to Wu Yang; yangwu@hrbeu.edu.cn

Received 23 May 2021; Accepted 9 August 2021; Published 15 September 2021

Academic Editor: Dr. Muhammad Shafiq

Information-Centric Networking with caching is a very promising future network architecture. The research on its cache deployment strategy is divided into three categories, namely, noncooperative cache, explicit collaboration cache, and implicit collaboration cache. Noncooperative caching can cause problems such as high content repetition rate in the web cache space. Explicit collaboration caching generally reflects the best caching effect but requires a lot of communication to satisfy the exchange of cache node information and depends on the controller to perform the calculation. On this basis, implicit cooperative caching can reduce the information exchange and calculation between cache nodes while maintaining a good caching effect. Therefore, this paper proposes an on-path implicit cooperative cache deployment method based on the dynamic LRU-K cache replacement strategy. This method evaluates the cache nodes based on their network location and state and selects the node with the best state value on the transmission path for caching. Each request will only select one or two nodes for caching on the request path to reduce the redundancy of the data. Simulation experiments show that the cache deployment method based on the state and location of the cache node can improve the hit rate and reduce the average request length.

1. Introduction

Information-Centric Networking (ICN) is a promising network architecture that is ideal for spreading popular information across the network. Its important feature is the node cache. A copy of the information is deployed to the cache of each router node. Each request to the router node is first searched in the content storage (CS). If there is a request with the same name in the CS, then the router respond with this copy directly. There is no need to fetch information from the content source server, thus reducing network traffic, reducing server bandwidth, and increasing the efficiency of content distribution. Under a reasonable cache deployment strategy and cache replacement strategy, caching will greatly improve the throughput and performance of ICN [1–3].

There are many cache nodes in the ICN. Excessive deployment of cache information on a tremendous number of nodes wastes a lot of storage space and causes data redundancy, which reduces the cache hit rate. Therefore, a good caching deployment strategy is needed to determine which node the cache is deployed. At present, the cache deployment strategy is mainly divided into a noncooperative cache and collaborative cache, and the collaborative cache is divided into an explicit collaboration cache and an implicit collaboration cache. Because in the noncooperative caching strategy, each node does not refer to the state of other nodes in the caching decision process and independently decides whether to cache content. Noncooperative caching will result in a large amount of redundant data in the network cache space, which is of little significance to the limited and valuable cache space [4]. Explicit collaboration caching generally requires a large amount of communication between the cache nodes to determine the best nodes to cache. However, this ignores the impact of network communication and computation on the nodes. The overhead data communication and calculation will increase the load on the network, which will reduce the overall performance [5–7]. In the implicit collaboration caching, the nodes incurred less overhead during the information exchange of the caching decision. Therefore, among noncooperative caching, explicit collaboration caching, and implicit collaboration caching, implicit collaboration caching has a very high practical significance, is not constrained by the specific circumstances of the network, does not increase excessive network traffic, and reduces caching redundancy [8]. For this reason, this manuscript proposes an on-path implicit cooperative cache deployment method based on the dynamic LRU-K cache replacement strategy. The main contributions of this manuscript are as follows:

(1) Through the dynamic LRU-K strategy, the state of the cache node and the network location are comprehensively judged

(2) Based on the state of the cache node and the location of the cache node in the network, a cache decision-making scheme is proposed in combination with implicit cooperative cache

(3) Improved multipoint caching and decision-making, reducing the repetition of network cache space content, while effectively increasing the cache hit rate

2. Related Work

Caching function is the advantage of ICN. The efficiency of caching directly determines the overall performance of ICN, and the caching deployment strategy of caching is the most important part of the caching strategies. In recent years, scholars around the world have focused more on the performance improvement of ICN brought by cache deployment strategy. For cache deployment strategy, we should focus on efficient network utilization and high availability of data. From the perspective of the P2P system and CDN technology, cache deployment on the edge of the network is very helpful to improve the cache hit rate, but ICN supports the deployment of caches on all routers. Not only the edge nodes but also the central nodes are suitable for the deployment of caches. Therefore, the deployment strategy of caches will be the key factor to increase the overall performance of ICN.

Cache deployment strategy is also called cache decision strategy. At present, there are two classifications of ICN cache deployment strategies in academia. One is classified according to the cache location, which is divided into an on-path strategy and an off-path strategy [9]. On-path strategy means that the cache is deployed on the sending path of interest packages. The specific cache nodes are determined by the cache deployment strategy [10, 11]. The off-path strategy does not depend on the sending path of the interest packet. It determines where the content should exist based on the overall network status or content attributes. The caching method related to the hash method is a typical path-independent caching method [12, 13]. Therefore, the off-path strategy is more suitable for the mobile cache. Another classification method is based on the cooperative approach, which is divided into three categories: noncooperative cache, explicit cooperative cache, and implicit cooperative cache.

The noncooperative caching strategy does not need to rely on the information of other caching nodes and can directly decide whether to cache the content or not. It is a strategy of "single-handedly fighting." This decision-making strategy is relatively simple and does not require excessive judgment. However, the problem that comes with it is that for an ICN with a holistic layout, such a simple cache strategy can affect the overall performance of the network. A typical caching strategy in a noncooperative cache is LCE [14]. In LCE, as the content information is returned, a cache is copied to each cache node passing through the return path. Although such a deployment strategy can simply cache the information content, it also brings high redundancy. The problem of a low overall hit rate has caused a serious waste of resources [15].

The explicit collaborative caching strategy determines the specific cache nodes by adding a centralized controller to the ICN. The centralized controller collects the real-time status of the cache nodes in the network and then makes a judgment. The presence of a centralized controller effectively relieves the pressure on the router. Studies have shown that explicit collaborative caching can greatly improve cache efficiency [16, 17].

In [18], an explicit cache coordination strategy based on HASH is proposed. The HASH value is obtained according to the location, popularity, and history of the cache node. Then, the corresponding cache node is searched for the cache. If this content exists in the cache node, the cached content is sent directly back to the subscriber, and if it does not exist, the subscriber's request is forwarded from this cache node. The HASH-based explicit cache collaboration strategy can quickly locate cache nodes and fast forward requests without generating redundant data.

Currently, the explicit collaboration cache is still under exploration. Although it can project a very good cache deployment strategy, it also has some shortcomings [19]. For example, in a high-speed network environment, each data content needs to pass the controller to help the controller decide which node to cache in; this computing overhead should not be underestimated. Secondly, if the centralized controller is down, it is unfavourable for the network where the centralized controller is located. Whether it hands off the cache node to another centralized controller or discards the current network, there is still a significant impact on overall ICN network.

Although explicit collaboration cache can bring high cache efficiency, it requires a lot of interactive information, and the calculation method is too complicated. The noncooperative cache has a lot of data redundancy. The implicit collaboration cache combines the advantages of the above two cache methods. The implicit collaboration cache mainly relies on some additional messages for cache decisions, such as cache node location, content popularity, probability, and cache node status. Due to the high performance and low cost of implicit collaborative caching, the implicit collaborative cache has the highest proportion among the three cache deployment strategies. Below are a few implicit collaboration caching strategies.

The main purpose of the LCD [20] (Leave Copy Down) strategy is to keep the cache close to the edge of the network, so the LCD policy will copy a cache on the second cache node of the return path each time, so each cache hit will move this cache once towards the edge node. As the content popularity increases, the cache will get closer and closer to the edge of the network, thus reducing the delay of user access. However, such a caching strategy will leave more copies on the return path with high redundancy. Moreover, for a large number of requests for different information, the cache of the core network nodes will be continuously replaced, and it is not guaranteed the cache hit.

MCD [21] (Move Copy Down) is an optimization of the LCD cache decision strategy. Unlike the LCD, after the cache hit, the cache of the node is deleted, and the redundancy of the information content is reduced. From the perspective of redundancy, MCD has made a very good optimization compared with LCD, but it also has certain disadvantages. Suppose there is a tree network, the root node is a content publisher, the leaf node is a content subscriber, and the other nodes are cache routers [22]. In this tree structure, if a router's cache is frequently hit by a request in a different subtree, the cached location will always be near this node and will not always go to the leaf node.

Prob [23] (Copy with Probability) is a random caching algorithm, sometimes called Bernoulli random caching algorithm, which uses a fixed probability p to make cache decisions. Such a caching algorithm does not guarantee that popular information will be cached. Popular content will have multiple requests, cached with the same probability, and will have more chances to be cached; this algorithm has a strong randomness.

ProbCache [23] proposed improvements based on Prob, and its improvement idea also hopes that the cache exists at the network edge as much as possible. Therefore, its probability is proportional to the distance from the cache node to the content provider (may be a content publisher or a cache node). The number of hops is used here as a measure of distance.

The above implicit cooperative caching schemes effectively reduce the burden of node information exchange of explicit cooperative caching, but they lack the consideration of the network node status and the network location of the caching node. The performance improvement of the caching is limited. In response to the above problems, this manuscript uses the LRU-K strategy to comprehensively judge the state and network location of the cache node. On this basis, this research combines the idea of implicit cache cooperation and proposes a cache decision-making scheme.

3. Analytical Methods

The research on ICN caching strategy can be classified into cache replacement strategy and cache deployment strategy. An effective cache replacement strategy can increase the hit rate of cache nodes. But for global purposes, reducing the average request length and cache redundancy requires a cache deployment policy to manage the global cache nodes [24].

According to the foregoing, implicit collaborative caching is a caching method that is very suitable for ICN. It does not require the global computing and communication capabilities of explicit collaborative caching, nor does it have the large amount of redundant data that noncollaborative caching generates [25]. Typical representatives in implicit collaboration cache are LCD [20], MCD [21], ProbCache [23], etc. These three caching strategies are designed to make the cache closer to the requesting node on the path, thus reducing network latency, but without considering each cache whether the state of the node is suitable to cache the information. Therefore, this paper proposes an on-path implicit cooperative caching algorithm with the following specific objectives:

(1) Improve the cache hit rate, reduce network latency, and reduce the average request length of users

(2) Consider the state and location of the cache node to give the nodes that should be cached

(3) Do not add too many fields to the packet, causing the network packet to be bloated

Based on the above requirements, this paper proposes a cache deployment method based on the status and location of the cache nodes, which considers the position of the cache node on the path, the number of prefiltering queues, and the number of filtering when caching nodes exchange information. It is a bidding strategy that determines the cache location at a very small communication cost [26].

3.1. Concept and Definitions

3.1.1. Cache Node Status Values. The cache node status value is an important indicator for evaluating whether the cache node in the ICN is suitable for caching. The less the cached content is, the more suitable it is the node for the cache. The closer to the user, the more suitable it is the node for the cache. At present, the state value of each cache node is determined by three factors. However, for the cache nodes in different network environments, the proportion of each factor should be different. Therefore, the final state value is calculated by weighting. The formula is as follows:

$$\text{Value} = \alpha \times V_k + \beta \times V_{\text{hop}} + \gamma \times V_{\text{hitk}}, \alpha, \beta, \gamma \in N. \quad (1)$$

Among them, Value represents the final state value, V_k represents the state value based on the number of prefiltered queues K, V_{hop} represents the state value based on the link location, V_{hitk} represents the data of interest packages in the number of prefiltered queues, and alpha, beta, and gamma represent the weight of three state values, respectively. In order to reduce the accuracy problem caused by floating-point representation, the weights of the three state values (α, β, γ) are represented by nonnegative integers.

(1) LRU-K Strategy. The main purpose of LRU-K is to solve the "cache pollution" problem of the LRU algorithm. Its core idea is to extend the "recently used 1 time" criterion to "recently used K times." The K in LRU-K represents the number of recent uses.

Compared with LRU, LRU-K needs to maintain one more queue to record the history of all cached data being accessed. Only when the number of accesses of the data reaches K times, the data is put into the cache. When data needs to be eliminated, LRU-K will eliminate the data whose Kth access time is the largest from the current time.

(2) Cache Queue State. The state of the cache queue is determined by the K value in the dynamic LRU-K algorithm, that is, the number of prefiltered queues. When the number of prefiltering queues of a cache node is large, it means that the number of packets need to be cached by the node is large. Compared with the cache node with a smaller K value, it takes a longer time to put the cache node with a larger K value into the cache queue. Therefore, the function for caching queue state calculation is a monotonic decreasing function.

(3) Link Location. In ICN, it is more desirable that information be cached on edge nodes, which can reduce the average number of request hops for users, the network latency, and the load of the central network. Because the acquisition of the whole network's topology structure is complex and requires a large amount of computation, it does not meet the requirements of implicit collaborative caching, so the hops of interest packages are added to the interest packages as the basis for calculating the relative location of cache nodes in the network. For example, the number of hops of the nearest cache node to the user is 1. If the cache node has no data in the CS, the cache node needs to forward the user's interest package to the cache node with the hop number of 2 until the content publisher or the node with the cache is found. The function of calculating the relative position state value of cache nodes by hops decreases monotonously with the increase of hops, and the higher the hops, the lower the importance.

(4) Location of Prefiltered Queues. In dynamic LRU-K cache replacement strategy, a data packet is put into the cache queue only after the packet has passed the required number of caches in the Kth queue at that cache node. Therefore, when judging whether a cache node is suitable for caching a packet, it should consider whether it is currently in the prefilter queue and which prefilter queue. Otherwise, the packet will appear in the prefilter queue of each cache node, thus underestimating the prevalence of the packet and delaying it from being cached or even failing to be cached. The purpose of this parameter is to reduce the impact of multiple nodes on data filtering and depending on the weight parameter that determines its importance in formula (1); priority should generally be given to nodes where that data is already in the prefilter queue so that it is cached as soon as possible.

3.1.2. Cache Rate. Caching rate is the ratio of the number of caches selected by the node to the number of interest packages received. The concept of caching rate is proposed mainly because when choosing the caching node through the state value, the edge nodes will be frequently selected as the caching nodes. The main reason for this problem is that the distribution of the state value is relatively uneven, and the caching probability of the nodes close to the content source is very small. Therefore, the cache rate is proposed, and the point on the forwarding path of the packet of interest with the smallest cache rate is selected for caching to compensate for the deficiency caused by selecting the cache node only by the state value. The formula for calculating the cache rate is as follows:

$$\text{CacheRate} = \frac{N_{\text{Cached}}}{N_{\text{total}}}. \tag{2}$$

In formula (2), N_{Cached} represents the number of data packets stored in the network cache space. N_{total} represents the number of all data packets in the network space.

3.1.3. Data Packet Status Value. A data packet is a response packet sent back by a content publisher according to the direction of the sending path of interest packages. The data package contains the name and content of the data. However, for the current deployment strategy, it is impossible to know whether the data packages have been cached on the path. Therefore, an additional field needs to be added to tell the subsequent caching node whether the data packages can be cached. When a data packet is cached at a node, the state value of the field should be modified, and subsequent nodes can judge whether to cache the data packet according to the state value. Two nodes, one has the maximum state value, and the other one has a minimum cache rate, need to be found on the transmission path of the packet. There are four cases, which can be represented by 0, 1, 2, and 3, respectively, as shown in Table 1.

3.1.4. Statement Record Table. In order to record the maximum state value and the minimum cache rate of a cache node through which an interest package passes, a Statement Record Table (SRT) is added to each cache node, which contains the data name field and the state value in the interest package. When the data packet returns, firstly querying the state value corresponding to the data name in the SRT table, if there is no record, it shows that the state value is not higher than the maximum state value, and the cache rate is not lower than the minimum cache rate at that time when the interest packet is transmitted, so this node is not selected as the cache node.

The meaning of state value in the SRT table is like Table 2, but it does not need state value 0. If it is not a candidate node of cache nodes, it cannot be stored in SRT. It is not necessary to prepare a state value specifically for this state, and it also reduces the occupied data space, the meaning of state value as shown in Table 2.

For the state value stored in SRT can not directly determine whether it is the maximum state value node and the

Table 1: Data packet status values and implications.

State value	Meaning
0	The maximum state value node has been cached, and the minimum cache rate node has been cached.
1	The maximum state value node is cached, and the minimum cache rate node is not cached.
2	The maximum state value node is not cached, and the minimum cache rate node is cached.
3	The maximum state value node is not cached, and the minimum cache rate node is not cached.

Table 2: State record table state values and implications.

State value	Meaning
1	It cannot be the maximum state value node, it may be the minimum cache rate node.
2	It may be the maximum state value node, not the minimum cache rate node.
3	It may be the maximum state value node or the minimum cache rate node.

```
Input: Interest Packet ($Pkt$); Statement Record Table ($SRT$)
Output: Operation Statement
$Pkt.hop \longleftarrow Pkt.hop + 1$
$value \longleftarrow$ get value
$cacherate \longleftarrow$ get cache rate
$srtstat \longleftarrow 0$
if $value > pkt.maxvalue$ then
    $pkt.maxvalue \longleftarrow value$
    $srtstat \longleftarrow srtstat \mid 2$
end if
if $cacherate < pkt.mincacherate$ then
    $pkt.mincacherate \longleftarrow cacherate$
    $srtstat \longleftarrow srtstat \mid 1$
end if
if $srtstat > 0$ then
    insert $pkt.name, srtstat$ into $SRT$
end if
forward $pkt$
return $SUCCESS$
```

Algorithm 1: Interest packet processing algorithm for cache deployment policy.

minimum cache rate node, it needs to be judged by combining the state value in the interest packet. The main reason is that in the process of forwarding the interest package, it is not known whether the next node has a larger state value or a smaller cache rate, but when the interest package is forwarded to the current node, it is the maximum state value or the minimum. When the data packet returns along the route forwarded by the interest packet, if a node encounters the maximum state value, and the state value in the data packet indicates that the node has not been cached at the maximum state value, then the current node is the node with the maximum state value, and the judgment of the node with the minimum cache rate is the same.

3.1.5. Concrete Design. The cache deployment strategy based on the state and location of the cache node selects two nodes to cache in the data transmission path. The first cache node is the one with the best state value, and the second cache node is the node with the lowest cache rate. At present, three factors are affecting the state value. The first one is the status of the LRU prefilter queue, which needs to be given in combination with the dynamic LRU-K algorithm proposed above. The second one is to calculate the location of each cache node on the request path according to the hops of interest packets. The closer to the edge of the network, the higher this value is. The third one is the queue number hit in the prefilter queue. The larger the number is, the more popular the content is, and the more important it is to cache the data packet at this node. Caching on the nodes with the lowest cache rate is to compensate for the nonuniformity of caching based on the state value.

As shown in Algorithm 1, in order not to add additional computing nodes when processing interest packets, the maximum state value and minimum cache rate of the cache nodes are allocated to the interest packets. Then, the state value and cache rate of the current nodes are calculated on each cache node. The larger state value and the smaller replacement rate are recorded in the

```
Input: Data Packet ($Pkt$); Statement Record Table ($SRT$); Content Store ($CS$)
Output: Operation Statement
if $pkt.datastat > 0$ then
    if $SRT$ has $pkt.name$ then
        $res \longleftarrow pkt.datastat \& SRT.getStat(pkt.name)$
        if $res > 0$ then
            $pkt.datastat \longleftarrow pkt.datastat - res$
            insert $pkt$ into $CS$
        end if
        remove $pkt.name$ from $SRT$
    end if
end if
forward $pkt$
return $SUCCESS$
```

Algorithm 2: Data packet processing algorithm for cache deployment policy.

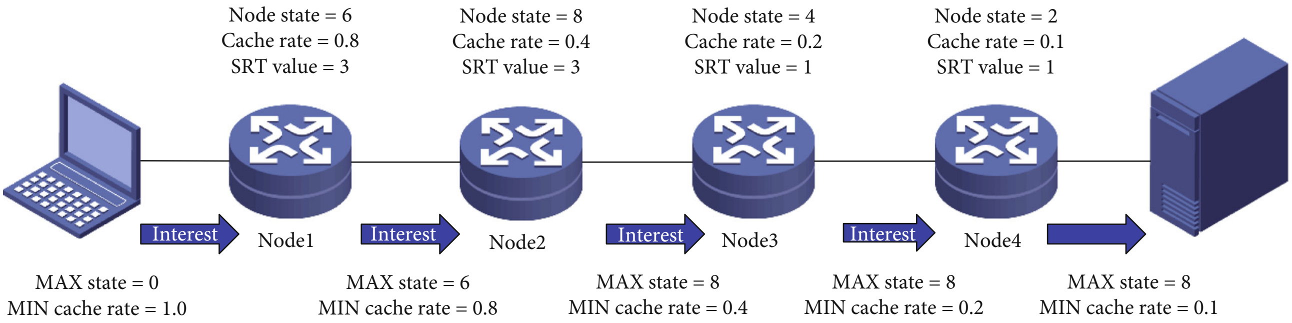


Figure 1: Interest packet forwarding process.

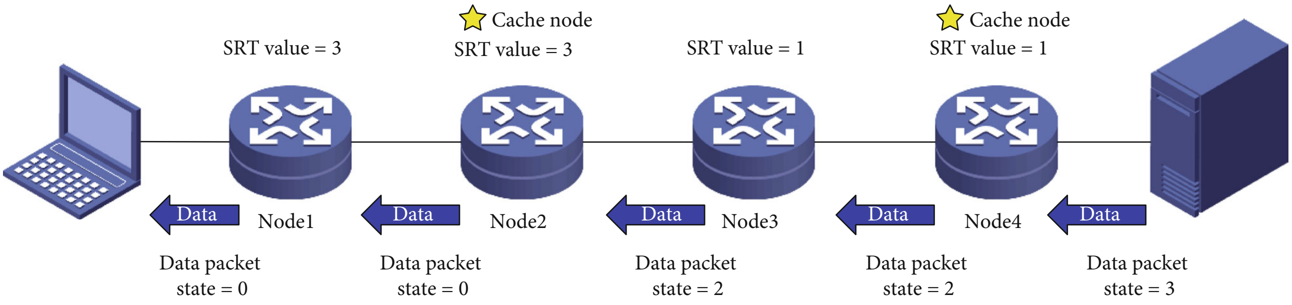


Figure 2: Data packet forwarding process.

interest packages and then insert data name and SRT status value into SRT. A maximum state value field, a minimum cache rate field, and a hop number field need to be added to the interest package. All cache nodes maintain these three fields together. If the state value of a cache node is higher than the maximum state value or the cache rate is lower than the minimum cache rate, the corresponding data is updated, and records are generated in the cache state record table, SRT state values are updated, and the cache state record table SRT is stored on each routing node.

As shown in Algorithm 2, when the cache node receives the response packet, whether the current packet is cached is calculated according to the SRT state value and the packet state value. When the content needs to be cached, the state value is changed so that the subsequent nodes will not cache the packet with the same type, thus ensuring that the data will not be excessively

Table 3: Icarus simulation platform parameters.

Parameter name	Parameter value
Zipf distribution alpha value	0.5-1.2
Number of contents	300000
Total cache size	(0.002-1) * number of contents
Interest packet preheat value	300000
Interest packet measurement	600000

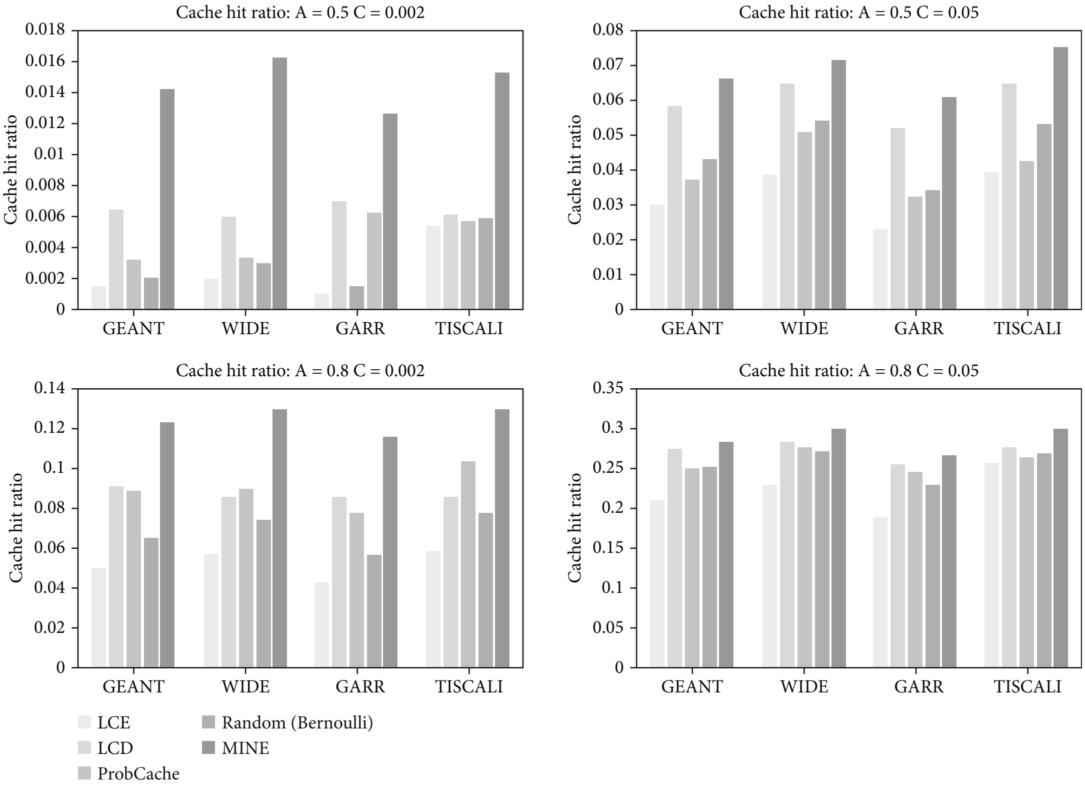


Figure 3: Comparison of cache hit rates.

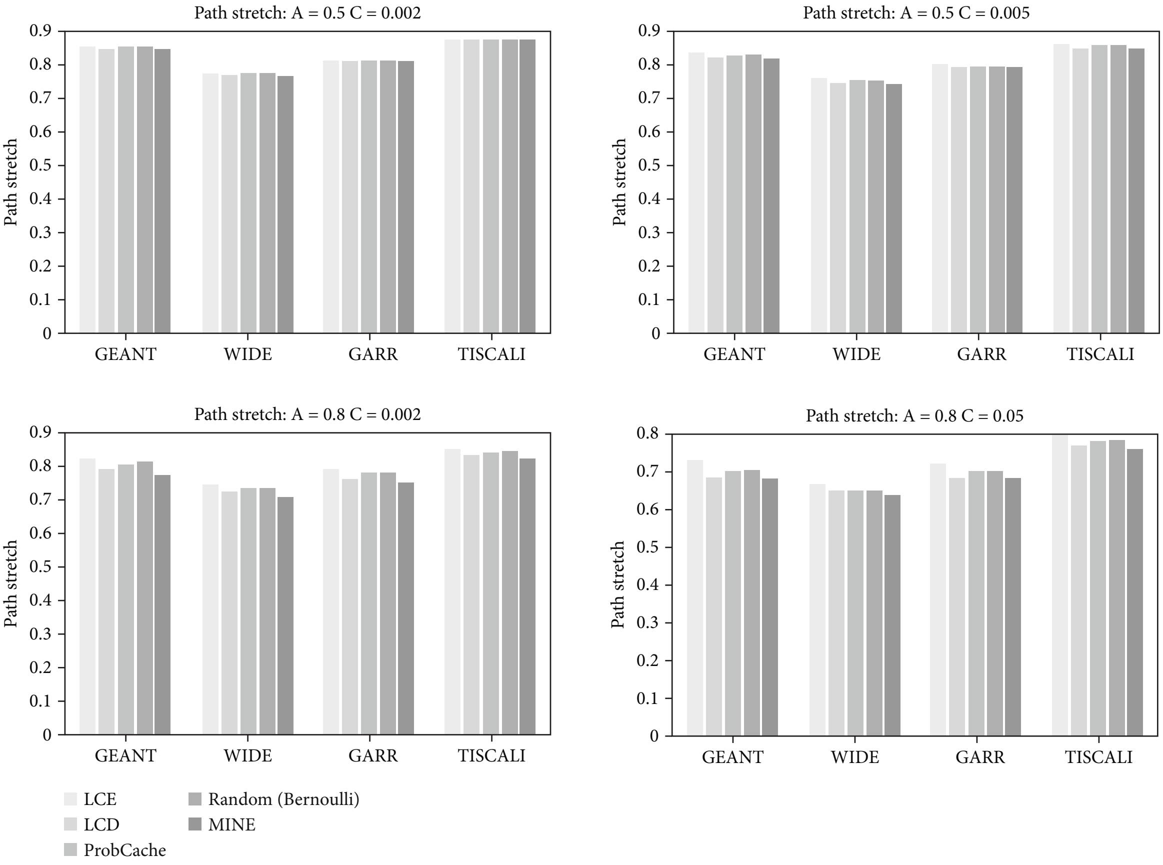


Figure 4: Path stretch comparison diagram.

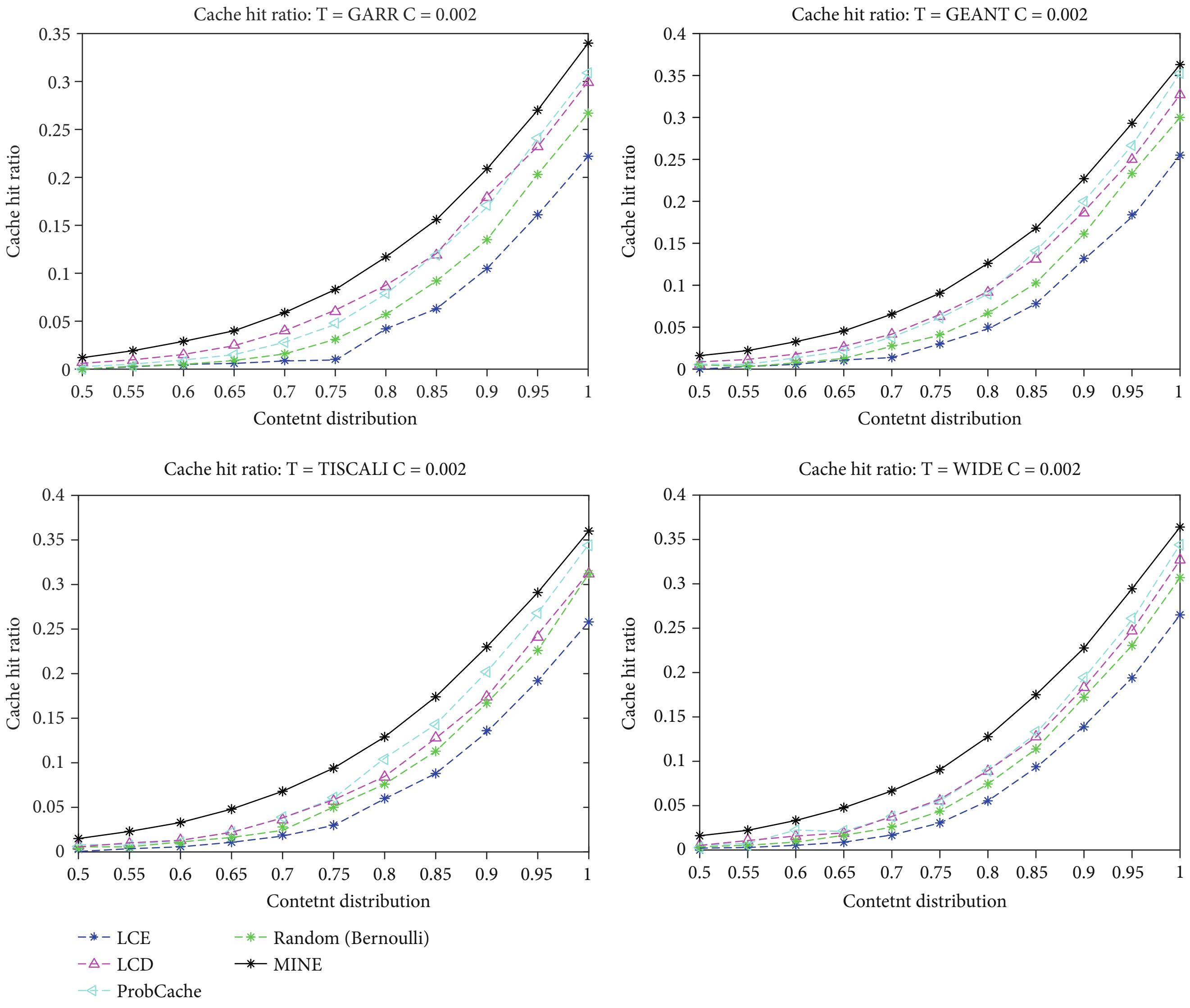


Figure 5: $C = 0.002$. The relationship between cache hit rate and content concentration.

redundant. If no caching is required, the data packet is forwarded directly. After the data packet passes through the cache node, the corresponding records in SRT should be deleted to avoid wasting storage space. In order to cooperate with the implementation of the algorithm, it is necessary to modify the data package accordingly. Add a packet status value field to the packet, the initial value sets the status to 3. In the return path, if the state value of the state record table in a node and the state value of a packet do not result in 1 when doing a logical and operation, then the cache is inserted into the cache of the node, and the state value of the packet is updated. If replacement is required, a dynamic LRU-K cache replacement strategy is executed.

The selection process of cache nodes is shown in Figures 1 and 2. Figure 1 shows the changes of some parameters in the process of forwarding interest packages. Figure 2 shows the selection of cache nodes in the process of packet response. Two graphs show the process of selecting cache nodes at one time.

4. Simulation Results and Analysis

4.1. Simulation Environment. The simulation environment uses the Icarus simulator, version v0.7.0, which can be downloaded from GitHub. The address is detailed in [27]. Icarus is an event-based ICN simulator, implemented in Python by Lorenzo Cerno and others. It is specially developed to evaluate the performance of the cache system in the information-centric network. It uses the MVC (Model-View-Controller) design pattern. This model implements the basic functions of ICN, the view monitors changes on the network model, the controller processes events and responses, and the results act on the network model.

The server used in the simulation experiment is the Sugon A620r-G server. The CPU is AMD Opteron(tm) Processor 6320, 1.4 GHz, 16 core, 32 threads, and 16 GB of memory, and the operating system is CentOS Linux release 7.2.1511. The parameters for the dynamic LRU-K cache replacement policy are set as follows:

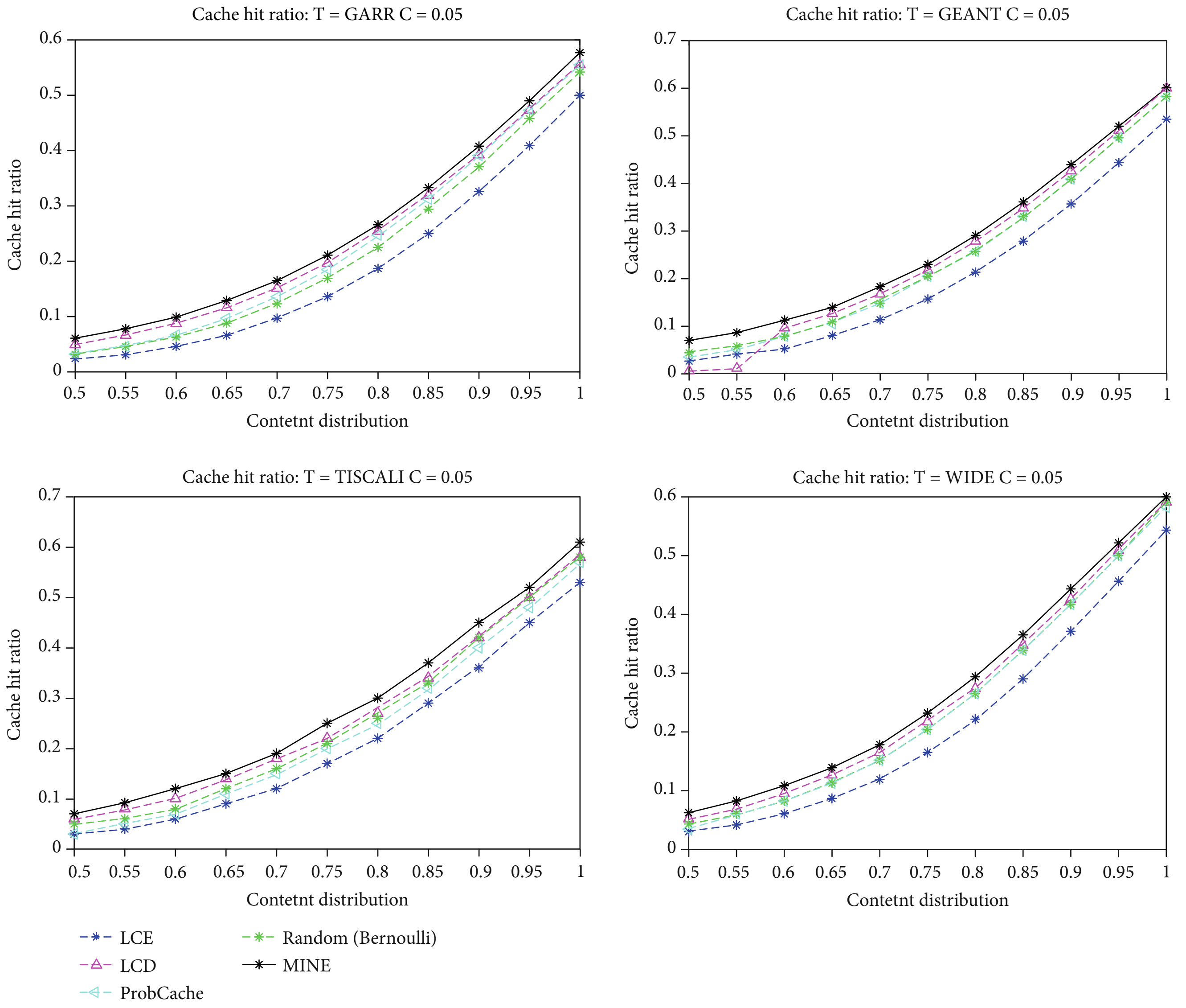


Figure 6: $C = 0.002$. The relationship between cache hit rate and content concentration.

(1) Prefilter queue length

$L(k) = L \times 2^{6-k}$, L is the length of the cache queue. The scheme used in the prefilter queue length in the simulation is an exponential function, which can effectively improve the hit rate.

(2) Prefilter queue number K maximum KMAX

$$\mathrm{KMAX} = 5. \tag{3}$$

(3) Filter count

The maximum number of filtering for the prefilter queue is 1.

The parameters for the Icarus simulation platform are set as Table 3. Since the initial state of the cache queue is empty, the cache replacement will not occur until the cache queue is filled, so it will affect the calculation of the hit rate. In order to eliminate this part of the impact, Icarus supports the warm-up function. The previous part does not collect data, and the data is collected for the hit rate after the number of requested packets exceeds the warm-up value, and the warm-up value can be freely set by the user.

4.2. Simulation Scheme. Icarus provides several topology data sets in Topology Zoo [28]. This simulation is tested in GEANT, GARR, TISCALI, and WIDE four topologies, with DLRU-K cache replacement strategy, FIFO cache replacement strategy, LRU cache replacement strategy, and RAND cache replacement strategy.

4.2.1. GEANT Topology Simulation Scheme. GEANT is a pan-European data network for research and education

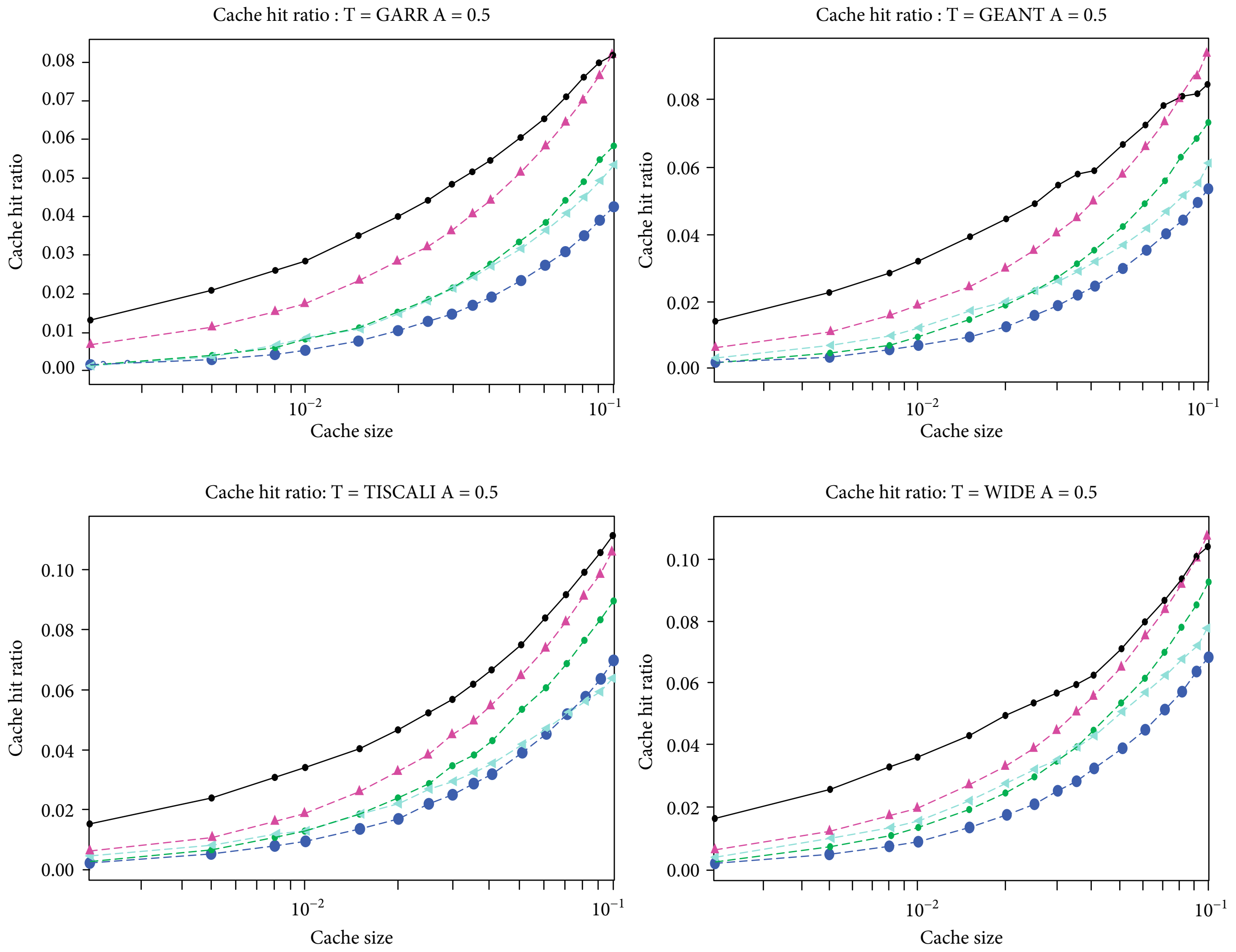


FIGURE 7: Relationship between cache hit ratio and cache capacity when $\alpha = 0.5$.

groups that connects research and education networks across Europe. In addition to providing high-bandwidth links in Europe, GEANT can also serve as a new technology test platform.

4.2.2. GARR Topology Simulation Scheme. GARR is a national computer network prepared for universities and research institutions in Italy. Its main goal is to design and manage a high-performance network infrastructure for the Italian academic and scientific communities. In fact, GARR has always been an integral part of the European research network GEANT, which shares GEANT with other European NRENs (national research and education network (NREN)). The GAR network topology provided by Icarus comes from Topology Zoo, and the version is GARR 201201.

4.2.3. WIDE Topology Simulation Scheme. Widely Integrated Distributed Environment (WIDE), an Internet project cofounded by Keio University, Tokyo Institute of Technology, and the University of Tokyo, is the backbone of the Japanese Internet and Japan's first Internet infrastructure.

4.2.4. TISCALI Topology Simulation Scheme. Since the real network topology of the real Internet service provider cannot be accessed by the research community, the Rocketfuel is used to map the nodes of the network. The TISCALI topology tested in this paper is mapped by Rocketfuel, and there are 44 content source nodes. There are 160 router nodes and 36 consumer nodes.

4.3. Simulation Result Analysis. The simulation results are mainly analyzed for the two performance indicators presented, the cache hit ratio and path scaling ratio under different network topologies. Cache hit ratio measures the response of cache nodes to interest packets in the network, which can intuitively reflect the efficiency of network cache deployment strategies. The path scaling ratio intuitively reflects the relative position of the cache deployment node on the path. The experimental results are as follows.

Figure 3 shows four cylindrical comparisons of the cache hit rates with different Zipf distribution parameters alpha and different cache sizes C. When the Zipf distribution parameters alpha = 0.5 and $C = 0.002$, the cache hit rates in the four network topologies have been significantly improved, which is higher than the experimental data of

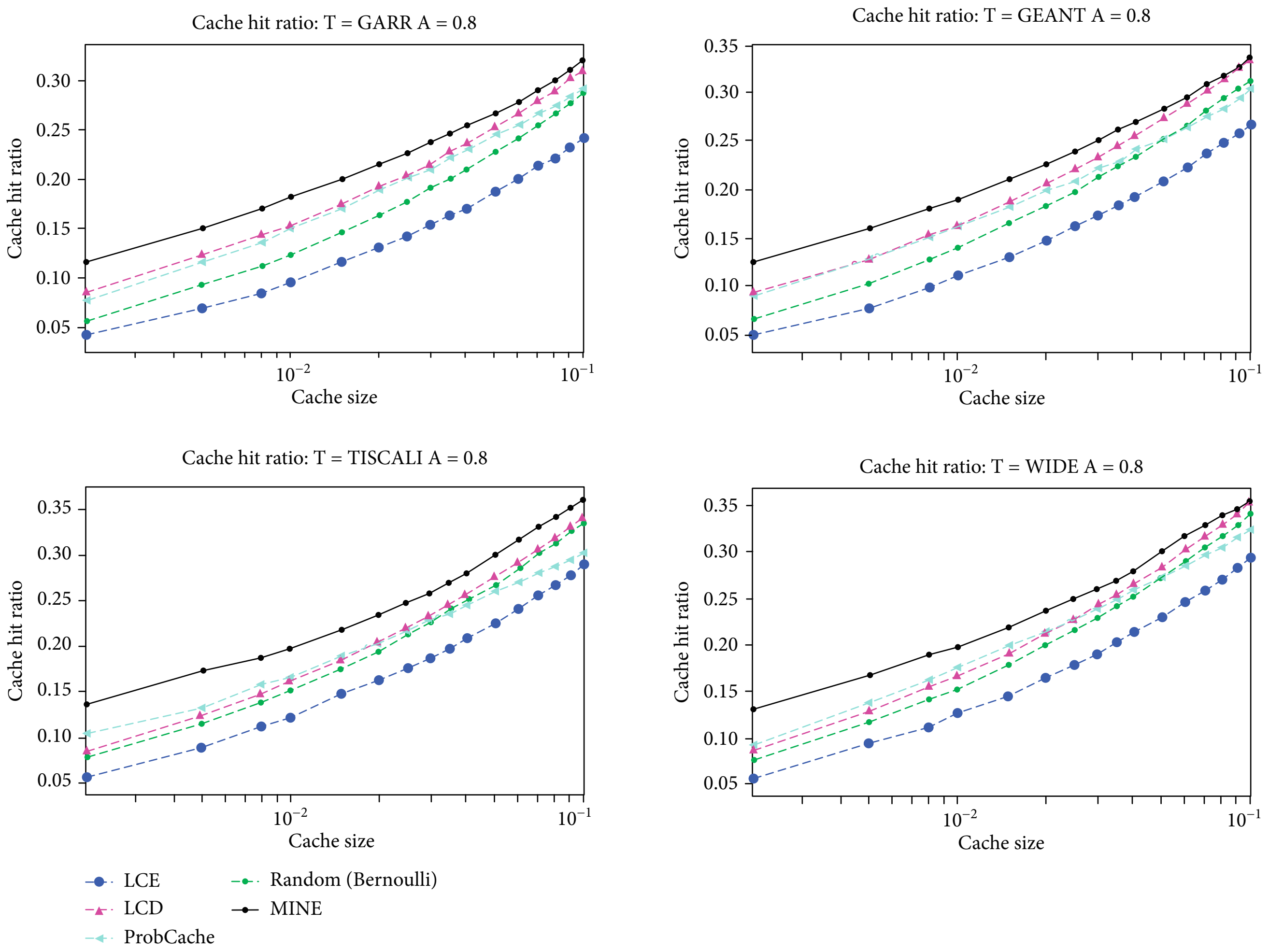


Figure 8: $\alpha = 0.8$, the relationship between cache hit rate and cache capacity.

the control group, and the fluctuation range of the hit rates is also smaller than the other four cache deployment strategies. The proposed caching strategy performs best among the five caching strategies, LCD is second only to the caching strategy in this paper, and the performance is relatively stable. However, the performance of ProbCache and Random in different network topologies is not stable and has certain volatility.

Compared with Figure 4, the path scaling ratio of the proposed cache strategy is also slightly lower than that of the other four control groups. By comparing the two groups of Figure 4, it can be found that the smaller the alpha and C, the more obvious the improvement of the cache hit rate is, but the reduction of path scaling ratio is not obvious. On the contrary, with the increase of alpha and C, the gap of path scaling ratio increases. The smaller the path scaling ratio is, the smaller the average request length is. Reducing the average request length not only brings faster response speed but also means better releasing the pressure of the core router so that interest packages can be responded at the edge of the network. For the forwarding process of each interest packet, even if the average number of hops is reduced by one hop, for the large number of interest packets existing in the network, it also means that the great load of the router is reduced, and the throughput of the router is increased.

The next part will mainly analyze the impact of different content popularity and cache size on the cache hit rate.

4.3.1. The Impact of Content Popularity on Cache Hit Rate. When $C = 0.002$ and $C = 0.05$, the break-line diagrams of cache hit rate are shown in Figures 5 and 6. The two break-line diagrams show that the cache strategy proposed in this paper is between alpha = 0.5 and alpha = 1.0. The cache hit rate in the four topologies is higher than that of other cache strategies. In GEANT and WIDE network topologies, when alpha reaches 1.0, the gap tends to decrease. With the increase of alpha value, the hit rate of all caching strategies increases, which indicates that the hit rate of caching increases with the increase of content concentration.

In the line graph, the cache hit rate of the five cache policies varies with the content popularity. The LCE cache deployment strategy has always been the worst performer. The hit ratio of LCD and ProbCache is closest to the cache strategy proposed in this paper. Even in the case of Figure 6, there is an almost equal situation. The caching strategy proposed in this paper gradually disappears when $\alpha = 1.0$.

4.3.2. The Effect of Cache Size on Cache Hit Ratio. The cache hit ratios when $\alpha = 0.5$ and $\alpha = 0.8$ are shown in Figures 7 and 8.

Through the two sets of broken-line graphs of Figures 7 and 8, it can be concluded that the cache hit rate of the four network topologies proposed in this paper is higher than that of other cache strategies when $C = 0.002$ to $C = 0.07$. However, when $C = 0.08$ and $C = 0.09$, the cache hit rate is almost equal to that of LCD strategy, and even when $a = 0.5$ and $C = 0.09$, the cache jitter occurs, which is surpassed by LCD strategy. The parameters of the scheme are not suitable for GEANT networks where C is greater than 0.08. According to other broken-line graphs, the cache hit rate of all caching strategies increases with the increase of caching, and the caching strategy awareness proposed in this paper is also getting smaller and smaller. According to the above analysis, the cache deployment strategy based on node status and location, combined with the dynamic LRU-K cache replacement strategy proposed in this paper, can improve the cache hit rate. The more decentralized the content and the smaller the cache, the more significant the enhancement effect.

5. Conclusions

ICN cache deployment strategy is divided into the noncooperative cache, implicit cooperative cache, and explicit cooperative cache. From the perspective of global cache hit rate and average request length, explicit cooperative caching generally has the best performance, but explicit cooperative caching requires a lot of communication information to determine the cached nodes, which will increase the network load. Therefore, this paper makes a thorough analysis and comparison of implicit cooperative caching. According to the advantages and disadvantages of other implicit cooperative caching, combined with the dynamic LRU-K cache replacement algorithm, a deployment strategy based on the status and location of caching nodes is proposed. According to the network location of caching nodes, the number of prefiltered queues, and the number of prefiltered times, the status is evaluated comprehensively, and the state value is put into interest. In the packet, the best state cache node is selected from the request path and cache in it when the packet returns. Finally, the deployment strategy based on the location and state of the cache node can reduce the duplication of the network cache space content and effectively improve the cache hit rates that are proved by the comparative simulation of the Icarus simulation environment.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research is supported by the National Natural Science Foundation of China (Grant Nos. 61771153, 61831007, and 61971154) and the Fundamental Research Funds for the Central Universities (Grant No. 3072020CF0601).

References

[1] W. X. Liu, J. Li, J. Cai et al., “Cod: caching on demand in information-centric networking,” *Telecommunication Systems*, vol. 69, no. 3, pp. 303–319, 2018.

[2] Y. B. Sun, Y. Zhang, and H. L. Zhang, “Survey of research on information-centric networking architecture,” *Acta Electronica Sinica*, vol. 44, no. 8, 2016.

[3] V. Jacobson, D. K. Smetters, and J. D. Thornton, “Networking named content,” in *Paper presented at the Proceedings of the 5th international conference on Emerging networking experiments and technologies*, 2009.

[4] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, “IoT malicious traffic identification using wrapper-based feature selection mechanisms,” *Computers & Security.*, vol. 94, p. 101863, 2020.

[5] L. Cai, J. K. Wang, and X. W. Wang, “Path cost and node cost based caching strategy for information-centric network,” *Journal of Chinese Computer Systems*, 2017.

[6] Y. Ding, Q. Zheng, and G. Chen, “Cooperative caching for ICN based on heat and cache replacement rate of node,” *Computer Engineering*, 2018.

[7] L. M. Huang, Y. Guan, and X. G. Zhang, “On-path collaborative in-network caching for information-centric networks,” in *Paper presented at the 2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2017.

[8] M. Shafiq, Z. Tian, Y. Sun, X. Du, and M. Guizani, “Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city,” *Future Generation Computer Systems.*, vol. 107, pp. 433–442, 2020.

[9] G. Q. Zhang, X. H. Wang, and Q. Gao, “A hybrid ICN cache coordination scheme based on role division between cache nodes,” in *Paper presented at the 2015 IEEE Global Communications Conference (GLOBECOM)*, 2015.

[10] S. Ioannidis and E. Yeh, “Adaptive caching networks with optimality guarantees,” *IEEE/ACM Transactions on Networking (TON)*, vol. 26, no. 2, pp. 737–750, 2018.

[11] H. Wu, J. Li, J. Zhi, Y. Ren, and L. Li, “Design and evaluation of probabilistic caching in information-centric networking,” *IEEE Access*, vol. 6, pp. 32754–32768, 2018.

[12] L. Saino, I. Psaras, and G. Pavlou, “Hash-routing schemes for information centric networking//ICN,” *ACM*, pp. 27–32, 2013.

[13] S. Wang, J. Bi, J. Wu, and A. V. Vasilakos, “CPHR: in-network caching for information-centric networking with partitioning and hash-routing,” *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2742–2755, 2015.

[14] N. Laoutaris, S. Syntila, and I. Stavrakakis, “Meta algorithms for hierarchical web caches,” in *Paper presented at the IEEE International Conference on Performance, Computing, and Communications*, 2004.

[15] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, “CorrAUC: a malicious bot-IoT traffic detection method in IoT network using machine learning techniques,” *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3242–3254, 2021.

[16] V. Pacifici and G. Dán, “Content-peering dynamics of autonomous caches in a content-centric network,” in *Paper presented at the 2013 Proceedings IEEE INFOCOM*, 2013.

[17] H. Wu, J. Li, T. Pan, and B. Liu, “A novel caching scheme for the backbone of named data networking,” in *Paper presented at the 2013 IEEE International Conference on Communications (ICC)*, 2013.

[18] L. Saino, I. Psaras, and G. Pavlou, “Hash-routing schemes for information centric networking,” in *Paper presented at the Proceedings of the 3rd ACM SIGCOMM workshop on Information-centric networking*, 2013.

[19] J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, and B. Fang, “A survey on access control in the age of Internet of Things,” *IEEE Internet of Things Journal.*, vol. 7, no. 6, pp. 4682–4696, 2020.

[20] C. Bernardini, T. Silverston, and O. Festor, “A comparison of caching strategies for content centric networking,” in *Paper presented at the 2015 IEEE Global Communications Conference (GLOBECOM)*, 2015.

[21] G. Q. Zhang, Y. Li, and T. Lin, “Caching in information centric networking: a survey,” *Computer Networks*, vol. 57, no. 16, pp. 3128–3141, 2013.

[22] M. Shafiq, Z. Tian, A. K. Bashir, A. Jolfaei, and X. Yu, “Data mining and machine learning methods for sustainable smart cities traffic classification: a survey,” *Sustainable Cities and Society*, vol. 60, p. 102177, 2020.

[23] I. Psaras, W. K. Chai, and G. Pavlou, “Probabilistic in-network caching for information-centric networks,” in *Paper presented at the Proceedings of the second edition of the ICN workshop on Information-centric networking*, 2012.

[24] J. Qiu, Y. Chai, and Z. Tian, “Automatic concept extraction based on semantic graphs from big data in smart city,” in *IEEE Transactions on Computational Social Systems*, vol. 7no. 1, pp. 225–233, 2020.

[25] Y. Wang, Z. Tian, Y. Sun, X. Du, and N. Guizani, “LocJury: an IBN-based location privacy preserving scheme for IoCV,” *IEEE Transactions on Intelligent Transportation Systems.*, vol. 22, 2020.

[26] A. S. Gill, L. D'Acunto, and K. Trichias, “Bidcache: auction-based in-network caching in ICN,” in *Paper presented at the 2016 IEEE Globecom Workshops (GC Wkshps)*, 2016.

[27] L. Saino, I. Psaras, and G. Pavlou, *Icarus: A Caching Simulator for Information Centric Networking (ICN)*, Paper presented at the SimuTools, 2014.

[28] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, “The internet topology zoo,” *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765–1775, 2011.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8736946, 9 pages
https://doi.org/10.1155/2021/8736946

Research Article

Binary Black-Box Adversarial Attacks with Evolutionary Learning against IoT Malware Detection

Fangwei Wang,[1,2] **Yuanyuan Lu**,[1] **Changguang Wang**,[1,2] **and Qingru Li**[1,2]

[1] *College of Computer & Cyber Security, Hebei Normal University, Shijiazhuang 050024, China*
[2] *Key Lab of Network & Information Security of Hebei Province, Shijiazhuang 050024, China*

Correspondence should be addressed to Changguang Wang; wangcg@hebtu.edu.cn and Qingru Li; liqingru2006@163.com

Received 28 May 2021; Accepted 12 August 2021; Published 30 August 2021

Academic Editor: Dr. Muhammad Shafiq

5G is about to open Pandora's box of security threats to the Internet of Things (IoT). Key technologies, such as network function virtualization and edge computing introduced by the 5G network, bring new security threats and risks to the Internet infrastructure. Therefore, higher detection and defense against malware are required. Nowadays, deep learning (DL) is widely used in malware detection. Recently, research has demonstrated that adversarial attacks have posed a hazard to DL-based models. The key issue of enhancing the antiattack performance of malware detection systems that are used to detect adversarial attacks is to generate effective adversarial samples. However, numerous existing methods to generate adversarial samples are manual feature extraction or using white-box models, which makes it not applicable in the actual scenarios. This paper presents an effective binary manipulation-based attack framework, which generates adversarial samples with an evolutionary learning algorithm. The framework chooses some appropriate action sequences to modify malicious samples. Thus, the modified malware can successfully circumvent the detection system. The evolutionary algorithm can adaptively simplify the modification actions and make the adversarial sample more targeted. Our approach can efficiently generate adversarial samples without human intervention. The generated adversarial samples can effectively combat DL-based malware detection models while preserving the consistency of the executable and malicious behavior of the original malware samples. We apply the generated adversarial samples to attack the detection engines of VirusTotal. Experimental results illustrate that the adversarial samples generated by our method reach an evasion success rate of 47.8%, which outperforms other attack methods. By adding adversarial samples in the training process, the MalConv network is retrained. We show that the detection accuracy is improved by 10.3%.

1. Introduction

With the commercialization and popularization of 5G, the IoT is coming closer to reality [1]. Meanwhile, with the scale expansion of connected terminals, data storage, and utilization, security issues are becoming more and more complex. As the methods of network crime are also constantly updated, the probability of network attack is greatly increased, which is not conducive to protecting personal privacy [2]. Therefore, in the 5G era, designing a model with good robustness is an important issue.

At present, malware attacks remain as one of the most urgent security issues users facing. In the last decade, deep neural network-based malware detection has fulfilled remarkable achievements [3]. A growing community of researchers is attempting to apply deep learning to malware detection and classification tasks [4–9]. Saxe and Berlin [10] extracted the binary features of PE files, which are portable executable ones under Windows operation systems and utilized a four-layer feed-forward neural network to detect malware. Kalash et al. [11] transformed malware binaries into greyscale images and classified malware by the use of a CNN. The DL-based malicious detection and classification models are now widely used.

However, much recent work indicates that adversarial attacks can cause serious damage to deep neural networks [12–15]. Adversarial examples in computer vision applications have been widely proven. In malware detection, adversarial modifications often need minor changes to malicious binaries. Different from language and images, codes are discrete sequences, which means that the generation technique of adversarial samples in images cannot be transferred to the malware detection field. Moreover, a minor change in the sequence may result in its functionality be changed completely. For example, in a binary file, changing a single byte may lead to a completely ineffective bytecode or distinct functionality. Therefore, it remains a great challenge to implement practical black-box attacks on malware binary-based deep learning models. Recently, a series of research works have been done in adversarial attacks. Hu and Tan [16] proposed adversarial modification of feature vectors. However, the malware binaries were not modified actually. This method cannot guarantee that the modified feature vector can be converted to actual binaries. Moreover, it destroyed the format or affected the functionality of the malware. Anderson et al. [17] directly modified malware binaries to perform adversarial attacks. In theory, these methods cannot disrupt the original functionality of the malware. However, in practice, we have found that these seemingly reliable methods also damage the malicious functionality. During an adversarial attack, if the malicious functionality of the original sample is destroyed, the adversarial attack is invalid. Therefore, it is necessary to perform malicious functionality detection on generated adversarial samples. However, most of the previous work did not address this issue. Besides, some of the previous work was done in a white-box adversarial model [18–20]. The white-box adversarial model requires knowing malware classifier architecture, making their methods impractical in real network environments. Therefore, while retaining the primary malicious functionality of the binaries, it is a great challenge to implement a practical black-box attack on the malware detection model based on deep learning.

This paper proposes an evolutionary algorithm-based adversarial sample generation method. In our approach, the generated samples by rewriting the file structure and adding adversarial information evade successfully the malware detection model, while preserving the original behavior of PE files. We test 1000 PE samples in four popular antivirus software on VirusTotal, showing that the method proposed can generate adversarial samples in binary format. The contributions of this paper are highlighted as follows.

(1) This paper proposes a new method of generating adversarial samples by the use of the evolutionary algorithm, which can automatically generate valid adversarial samples

(2) This paper uses a well-designed feature library as rewriting material in the evolutionary process, which helps generate modified samples with fewer attempts

(3) This paper applies the adversarial samples generated to attack DL-based malware detection engines on VirusTotal and obtains better experimental results than other attack methods

The rest of this study is organized as follows. Section 2 is a concise introduction of malware detection and adversarial attack methods. Section 3 proposes our attack framework in detail. Section 4 describes the experimental settings and main results and gives a deep analysis. Section 5 concludes this paper as well as the research directions.

2. Related Work

2.1. Machine Learning-Based Malware Detection Methods. In malware detection, machine learning (ML) is a popular approach. Moreover, in recent years, many ML-based malware detection methods have been put forward [21–24]. These methods are mainly categorized in static analysis [25, 26] and dynamic behavior analysis [27, 28]. Static analysis learns the statistical features of malware (e.g., API calls, OpCode), whereas dynamic behavior analysis detects abnormal (possibly malicious) behavior by observing deviations from the baseline of the system. Recently, malware detection efforts prefer to use raw software binaries as the input of DL models [29–31].

NVIDIA's research group [32] proposed the MalConv network, which took the raw byte sequences of PE files as input directly, achieved 98.88% detection accuracy. Compared with the detection model that extracts only some features of PE files as input, MalConv links other discrete features. Therefore, it can detect samples with arbitrary size and avoid missing important features. This paper assesses the effectiveness and performance of our framework using the MalConv detection system.

2.2. Adversarial Attack against Malware Detection Model. DL-based malware detection approaches are susceptible to adversarial attacks [33–37]. Adversarial modifications by manipulating only a small fraction of raw binary data may lead to misclassification. Moreover, the raw binary contents of data are not changed in a nutshell; otherwise, its originally momentous functionality might lose.

Prior work has proposed various ways of adversarial attack against ML-based malware detection models. Through appending bytes at the end of a binary file while preserving its intrusive functionality, Kolosnjaji et al. [20] designed a gradient-based attack model. However, it is based on white-box attacks and cannot be applied to real scenarios. Kreuk et al. [38] proposed a modification method that injected a minor byte sequence into the originally binary file. It is also based on white-box attacks and is not efficient in real scenarios. Anderson et al. [17] designed an effective model which is based on a deep reinforcement learning method to attack static PE antimalware engines. In their work, the reward function and the environment of reinforcement learning were artificially defined. Later, Fang et al. [39] improved Anderson et al.'s work by autonomously generating the reward function according to the expert strategy. Numerous experiments showed that Fang et al.'s method [39] is more nimble and efficacious than Anderson et al.'s method [17]. Yuan et al. [40] proposed an adversarial sample generation model named GAPGAN. GAPGAN initially maps the discrete malware binaries into a contiguous space;

the output is input to the generator of GAPGAN to generate adversarial payloads. Finally, the generated payloads are appended to the originally binary file to create an adversarial one. Because the valid part of the binary file was not changed, the original functionality of the binary file is preserved. GAPGAN can perform an efficient black-box attack. However, the modification action in GAPGAN involves only a simple action. The GAPGAN cannot perform complicated modifications similar to real malware writers. Song et al. [41] presented a framework for creating adversarial malware and evaluated the evasion capabilities in realistic scenarios. The authors firstly revealed the root causes that adversarial samples evade the malware detection method.

This study puts forward a novel binary manipulation-based attack framework, which generates adversarial samples with an evolutionary learning algorithm. Our method can adaptively simplify the actions of modifying binary samples and use an evolutionary algorithm to make adversarial samples more targeted. The generated adversarial samples by statically rewriting the PE file keep their dynamic behavior consistent and can evade the DL-based malware detection models. Experimental results verify the effectiveness of our method, which can efficiently and quickly generate adversarial samples without human intervention.

3. Adversarial Sample Generation Based on Evolutionary Algorithm

3.1. Problem Description. Our ultimate objective is to generate an antagonistic sample, which manipulates the classifier to classify malicious software as benign by mistake while still retaining the malicious function of the original sample.

Let us consider a classifier f that maps any binary file S into a unique category label. Label = 0 denotes that S is malicious, and label = 1 denotes S is benign. Let $A = \{a_1, a_2, \cdots, a_n\}$ be an action set that is used to modify the malware samples. S denotes an original sample, whereas S_{mod} denotes a modified sample. The functionality detecting function v is used to check whether S_{mod} retains the same malicious functionality with S. When the output of the functionality detection function is 1, we consider that S_{mod} retains the malicious functionality of S and then save S_{mod} as the adversarial sample S_{adv}. The specific formulae can be detailed in Section 3.3.3.

In brief, for the malware sample S, our goal is to generate an adversarial sample S_{adv} which makes $f(S_{\text{adv}}) = 1$, and if $v(S, S_{\text{mod}}) = 1$, $S_{\text{adv}} = S_{\text{mod}}$.

3.2. Rewriting Actions. PE file is a generic term for executable files in Windows operating system. A PE file consists of a header, section table, and section data. The MS-DOS header consists of three parts: a DOS header, the true PE header, and an optional header, and it includes some basic messages about the executable file. Section table describes the characteristics of each file section. The section table consists of a series of IMAGE_SECTION_HEADER structures arranged in a sequence. The structures and sections are arranged in a fixed order. Section data consists of 4 main parts: .text, .data, .rdata, and .idata, and the data part includes the practical contents relating to every section. The PE file format is shown in Figure 1.

For a black-box model, we have no idea of the exact features of the classifier involved. However, by observing the chosen features in some open-sourced classifiers, we can make a wild guess at some of the common features in malware detection models. An adversarial sample is generated by modifying one or several features. The chosen actions of modifying the features should be easy to execute. Moreover, after the features are modified, the executability and functionality of malware should not be corrupted. In this paper, all actions applied to the PE file are shown as follows.

(1) Appending some bytes to the PE file

(2) Inserting an unused function to the import address table

(3) Appending some bytes to the untapped space in a section

(4) Adding a new section

(5) Changing a section name

(6) Packing the file

(7) Unpacking the file

The malicious binary file is modified through the following steps. Firstly, the original PE file is read, then the content in the specified location is added or deleted, and finally, the relative virtual address of the PE file is modified.

3.3. The Proposed Framework. The workflow of the framework includes three parts: the generation of the feature library, the generation of the modified samples, and the generation of the adversarial samples. Firstly, the feature library is generated using MalGAN. In the processing of generating the adversarial samples, the modified features are randomly selected from the feature library according to the rewriting actions. Then, modified samples by evolutionary algorithms are generated. Finally, the generated modified samples are tested whether the malicious functionality of the original samples remains or not. If a modified sample has the same malicious functionality as the original one, we save it as an adversarial sample. Figure 2 gives an overview of our framework. The details of the three parts are given in the following.

3.3.1. Generation of Feature Library. To efficiently generate adversarial samples, we collect and generate the rewritten feature library using MalGAN. MalGAN, proposed by Hu and Tan [16], is used to generate adversarial samples for attacks based on GAN. The MalGAN architecture primarily consists of three components: a generator, a discriminator, and a black-box detector. By only adding a random number to API calls, MalGAN can transform a malicious feature vector into its opposed version.

Our work is built on this work. Firstly, we construct the sample library consisting of malicious and benign samples and extract their binary features, such as the import

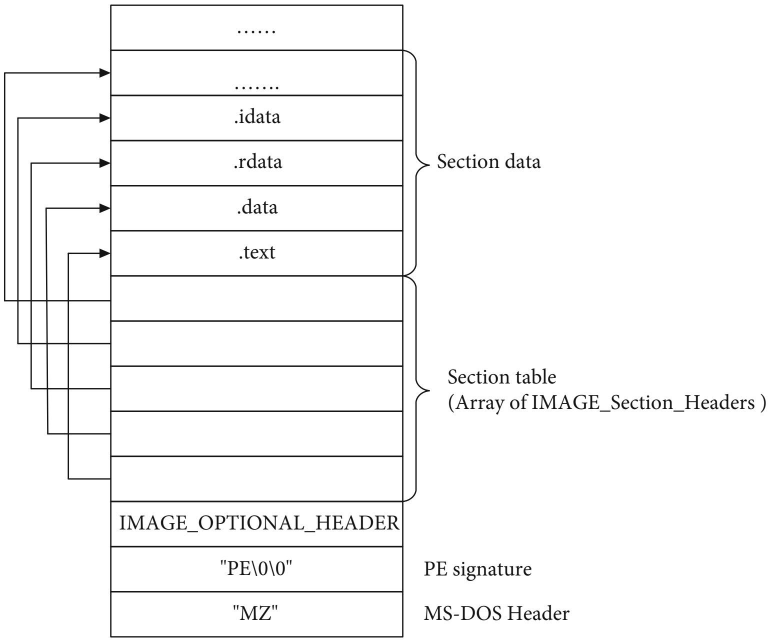


Figure 1: PE file format.

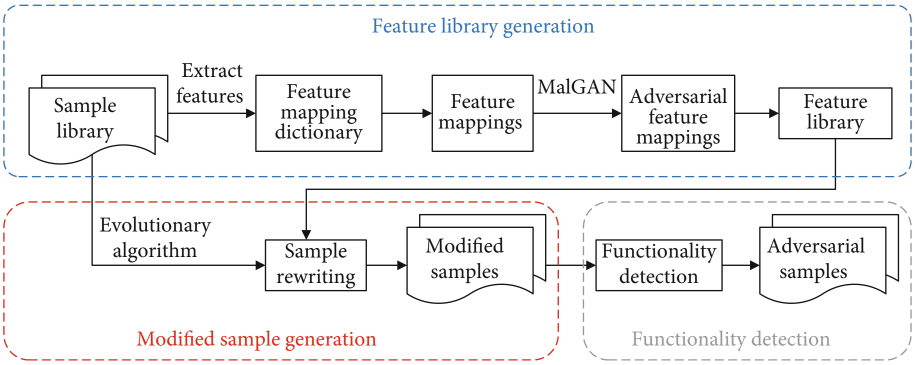


Figure 2: Overview of the framework.

functions and section names. Then, they are stored in a feature mapping dictionary for convenient retrieval and future operations. Next, we use the feature mapping dictionary to generate separate feature mapping for each malicious and benign sample and send them as the input to the MalGAN. After running for a few epochs, the MalGAN can generate adversarial feature mappings. Finally, according to the adversarial feature dictionary, the adversarial feature mappings are mapped into the feature library.

Once the feature library is generated, the modified features needed are randomly selected from the feature library according to the rewriting operation when generating modified samples.

3.3.2. Generation of Modified Samples Based on Evolutionary Algorithm. Evolutionary algorithms simulate the evolution of species in nature, such as selection, crossover, and mutation, which are often used to solve some optimization problems by choosing the brightest individual from the whole population. Different from traditional optimization algorithms such as calculus-based methods and exhaustive enumeration methods, evolutionary learning is a global optimization algorithm, which is highly robust and widely applicable.

This study uses an evolutionary algorithm for sample rewriting to generate modifications. The evolutionary algorithm can adaptively simplify the actions of modifying samples and make the adversarial sample more targeted. It can efficiently generate modified samples without human intervention. Compared with other existing methods, the evolutionary algorithm starts from the string set, which improves the speed of the algorithm and is easy to parallel computing. There is no backpropagation of weights and biases in deep learning and optimization of the loss function, which decreases the probability of obtaining a local optimum.

In the process of evolution, malware samples are considered individuals. Atomic manipulation of rewriting samples is a gene with a genetic message, and the predictive effect of the detecting model on modified samples is fitness. The generation process of the modified sample is shown in Figure 3.

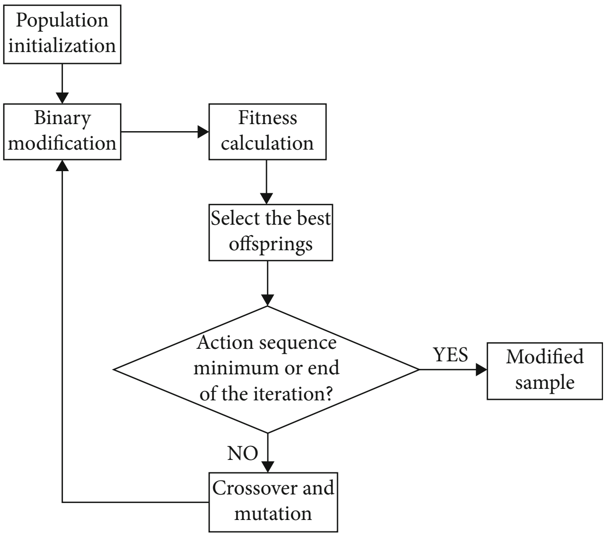

Population initialization
Binary modification
Fitness calculation
Select the best offsprings
Action sequence minimum or end of the iteration?
YES
Modified sample
NO
Crossover and mutation

Figure 3: The generation process of a modified sample.

The detailed evolutionary process is as follows.

(1) Step 1. Population initialization. n segments of the genome are generated randomly

(2) Step 2. Binary modification. Firstly, n segment genomes from binary sequences are mapped to candidate action sequences. Then, the malware samples are rewritten by candidate action sequences to generate modified samples

(3) Step 3. Fitness calculation. The modified samples are fed into the MalConv network, and the output of the MalConv is used as the fitness of the individual. A smaller output value of MalConv indicates a higher fitness. The higher the fitness, the higher the probability that a gene sequence will be selected for retention

(4) Step 4. Selecting the best offspring according to the fitness as the parent of the next generation

(5) Step 5. Performing genetic manipulation on the selected parents. New offspring through crossover and mutation are reproduced

(6) Step 6. The assessment of end condition. When the action sequence has reached a minimum value or the maximum number of iterations is reached, the evolution is ended. If the end condition is satisfied, the modified samples are output. Otherwise, skip to Step 2

The detailed process is described in Algorithm 1.

3.3.3. Generation of the Adversarial Samples. It should also be noted that the functionality of a malware sample may be corrupted during the modification process. In other words, its attacking characteristics may be damaged. We consider an adversarial sample without malicious functionality to be invalid. To detect whether the malicious functionality of the modified sample is retained, we use the sandbox to collect behaviors of the modified samples and original samples. If the behavior of the modified sample is the same as that of the original one, we think that it retains the malicious function of the original sample, and it is saved as an adversarial sample.

Suppose a behavior of the original sample S is indicated as the set B_S, and behavior of the modified sample S_{mod} is indicated as set B_{mod}. We denote the total number of similar behaviors in B_S and B_{mod} as $\mathrm{num}(B_S, B_{\mathrm{mod}})$ and the size of B_S as $\mathrm{num}(B_S)$. The behavior similarity between B_s and B_{mod} is defined as sm:

$$\mathrm{sm} = \frac{\mathrm{num}\left(B_S, B_{S_{\mathrm{mod}}}\right)}{\mathrm{num}(B_S)}, \tag{1}$$

$$v(S, S_{\mathrm{mod}}) = \begin{cases} 1, \mathrm{sm} \geq 0.7, \\ 0, \text{elsewhere}. \end{cases} \tag{2}$$

Because the modification operation is a direct manipulation on the original sample, it can inevitably alter the behaviors of the original samples. Therefore, we assume that if $\mathrm{sm} \geq 0.7$, the samples S and S_{mod} have the same behaviors; that is, $v(S, S_{\mathrm{mod}}) = 1$. It also means that the modified sample retains its original malicious functionality. In the end, we save the modified sample S_{mod} retained originally the malicious functionality as the adversarial sample S_{adv}.

4. Experimental Results and Analysis

This section firstly gives the setting of our experiments, including the datasets, evaluation metrics, and the target malware detection model. Then, we analyze the experiment results.

4.1. Experimental Settings and Evaluation Metrics. In the experiment, we construct a dataset with 1000 malware samples from VirusTotal. Moreover, we also produce some adversarial examples for PE binaries to evaluate the effectiveness of our proposed method.

To assess the effectiveness of adversarial samples, we measure some evaluation metrics in Table 1. N_s denotes the number of modified files that have structural integrity or executability. N denotes the total amount of samples. N_r denotes the number of modified samples that retain the originally malicious functionality. N_e denotes the number of adversarial ones that can evade malware detection engines. The computer specification used for the experiments is as follows: CPU: Intel Core I5-6500, 3.20 GHz, 4 cores, 8 threads; memory: 2 GiB; and operating system: Ubuntu 16.04.

4.2. Experimental Results of Adversarial Attack. This section demonstrates some performances of our approach under antiattack scenes and compares the results with some methods available.

In our experiment, the attacked model is the MalConv proposed by Raff et al. [32]. We train attacked MalConv network using a dataset with 6230 malicious samples from

```
Input: malware samples $S$, population scale, number of generations $G$.
Output: modified samples $S_{mod}$
BEGIN
for $s$ in $S$ do
    Initialize the population;
    while current generation $\leq G$ or action sequence is not minimum do
        Map binary sequences to action sequences;
        Modify malware sample based on the action sequences;
        Calculate fitness;
        Select the best offspring;
        Perform crossover;
        Perform mutation;
        Increase current generation;
    end while
    Append the optimal result $S$ to $S_{mod}$;
end for
Return $S_{mod}$;
END
```

Algorithm 1: Malware sample evolution.

Table 1: Evaluation metrics.

Evaluation metrics	Formula
The success rate of modified sample operation (R_s)	$R_s = N_s/N$
The malicious function retention rate of modified samples (R_r)	$R_r = N_r/N_s$
Evasion rate (R_e)	$R_e = N_e/N_r$

Table 2: The performance comparison of adversarial samples generated by different methods.

Attack methods	Evaluation metrics		
	R_s	R_r	Average time
Our method	100%	97.5%	24.5 s
DQEAF	76%	94.3%	60 s
Aut.	68%	95.6%	42.5 s

Table 3: The evasion rate of different attack methods against different detection engines.

Detector	Adversarial attack methods		
	DQEAF	Aut.	Our method
ClamAV	19.2%	17.3%	18.5%
Cylance	39.5%	42.5%	47.8%
Endgame	21.3%	22.6%	23.5%
Trapmine	20.6%	18.8%	19.2%

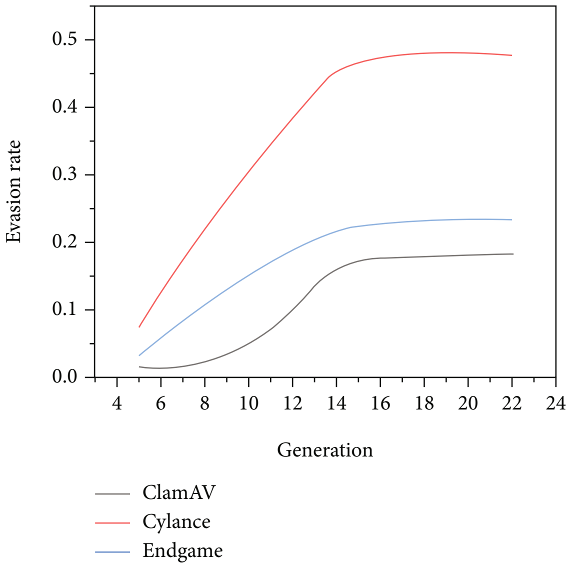


Figure 4: The evasion rate with the generation increase.

Table 4: Performance evaluation on MalConv with and without adversarial training.

Defense	Test dataset	Accuracy
No defense	×	98.4%
	√	80.2%
Adversarial training	√	90.5%

× indicates no adversarial samples and √ denotes having adversarial ones in the test dataset.

VirusTotal and 5660 benign samples from web crawling and achieve 98.4% detection accuracy.

Furthermore, to explore the validity of the presented binary-based attack method against a deep learning-based detection system, we compare our approach with other byte-level attack methods, including the DQEAF method [39], which is based on reinforcement learning and Aut. method [41], which is based on code randomization and binary manipulation. The results are shown in Tables 2 and 3, respectively.

Table 2 shows the performance comparison of adversarial samples generated by different methods. From the three evaluation metrics of generated samples, our approach outperforms other similar methods. The reason is that the selected action of modifying malicious samples is built on ensuring the execution of PE files. The specific modifying actions, such as inserting, changing, and adding actions, are taken from the generated feature library. The feature library has also been carefully designed to ensure the authenticity of all actions. Therefore, the modification does not involve invalid actions which damage the primary structure of malware or lose its original functionality. Meanwhile, our action set does not cover the irreversible actions, such as removing signatures, which makes our action set is more effective. Our approach can do a heuristic random search which simplifies the modification actions and does not require performing action sequence minimization and marking action weights and success content as the other two methods, which significantly raises the efficiency of our method.

To further test the effectiveness, we evaluate it using four representative malware detection engines on VirusTotal, including ClamAV, Cylance, Endgame, and Trapmine. Table 3 demonstrates different evasion rates of adversarial samples produced by some methods against different detection engines. From Table 3, we can see that the evasion rate of our method has a better performance compared with the other two in most cases. Meanwhile, to test the efficiency, we record the evasion rate with the generation increase, shown in Figure 4. From Figure 4, we can find that our method reaches a relatively stable evasion rate after 15 generations, which shows that our method is very efficient.

4.3. Defense against Adversarial Sample Attack. To defend against adversarial attacks, more and more defense countermeasures have been proposed. Among them, adversarial training is one of the most popular ways [42], in which adversarial samples are added to the training set; thus, DL models can adjust the decision strategies. Compared with other adversarial defense methods, adversarial training does not require modifying the detection model and is easy to implement.

In this paper, we use adversarial samples generated by evolutionary algorithms to test on MalConv network. Table 4 shows the performance evaluation on MalConv with and without adversarial training. The experimental results show that the detection accuracy increased from 80.2% to 90.5% after the adversarial training. It also illustrates that adversarial training can effectively improve the model robustness to adversarial attacks.

5. Conclusion

To make DL-based IoT malware detection models more robust and effective, we propose a framework for generating adversarial samples and their defense. Our framework firstly adopts an evolutionary algorithm to generate modified samples, and then, the modified samples that retain the originally malicious functionality are saved as adversarial samples. This method does not need to obtain any information of the special detection models containing extracted features, internal parameters, etc. Moreover, our approach is entirely automated without human intervention. The experimental results demonstrate that our method can ensure the diversity of generated samples and greatly enhance the efficiency of adversarial sample generation. This paper also demonstrates that adversarial training is one of the effective methods to combat adversarial sample attacks.

The action space has a great influence on the diversity and versatility of evolutionary optimization algorithms. Defining more effective modification actions to expand the search space of evolutionary algorithms is our urgent task. Our future work also includes accelerating the convergence speed and improving the stability of the evolutionary algorithm. Moreover, we will also explore more methods of generating adversarial samples to defend against adversarial attacks on IoT.

Data Availability

The dataset can be obtained from the website: https://www.virustotal.com (accessed on 22 April 2021).

Conflicts of Interest

The authors declare no conflict of financial or associative interest in connection with the manuscript submitted.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No. 61572170), the Natural Science Foundation of Hebei Province of China (Grant Nos. F2019205163 and F2021205004), the Science and Technology Foundation Project of Hebei Normal University (Grant No. L2021K06), the Science Foundation of Returned Overseas of Hebei Province of China (Grant No. C2020342), the Science Foundation of Department of Human Resources and Social Security of Hebei Province (Grant Nos. 201901028 and ZD2021062), and the Natural Science Foundation of Hebei Normal University (Grant No. L072018Z10).

References

[1] J. Qiu, Z. Tian, C. du, Q. Zuo, S. Su, and B. Fang, “A survey on access control in the age of Internet of Things,” *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4682–4696, 2020.

[2] Y. Xu, C. Zhang, Q. Zeng, G. Wang, J. Ren, and Y. Zhang, “Blockchain-enabled accountability mechanism against information leakage in vertical industry services,” *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1202–1213, 2021.

[3] X. Yan, Y. Xu, X. Xing, B. Cui, Z. Guo, and T. Guo, “Trustworthy network anomaly detection based on an adaptive learning rate and momentum in IIoT,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6182–6192, 2020.

[4] A. Sharma, P. Malacaria, and M. H. R. Khouzani, “Malware detection using 1-dimensional convolutional neural networks,” in *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pp. 247–256, Stockholm, Sweden, 2019.

[5] X. Pei, L. Yu, and S. Tian, "AMalNet: a deep learning framework based on graph convolutional networks for malware detection," *Computers & Security*, vol. 93, p. 101792, 2020.

[6] J. Hemalatha, S. A. Roseline, S. Geetha, S. Kadry, and R. Damaševičius, "An efficient densenet-based deep learning model for malware detection," *Entropy*, vol. 23, no. 3, pp. 344–367, 2021.

[7] S. Yoo, S. Kim, S. Kim, and B. B. Kang, "AI-HydRa: advanced hybrid approach using random forest and deep learning for malware classification," *Information Sciences*, vol. 546, no. 9, pp. 420–435, 2021.

[8] Y. Xu, J. Ren, Y. Zhang, C. Zhang, B. Shen, and Y. Zhang, "Blockchain empowered arbitrable data auditing scheme for network storage as a service," *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 289–300, 2019.

[9] Y. Xu, Q. Zeng, G. Wang, C. Zhang, J. Ren, and Y. Zhang, "An efficient privacy-enhanced attribute-based access control mechanism," *Concurrency & Computation Practice & Experience*, vol. 32, no. 5, 2020.

[10] J. Saxe and K. Berlin, "Deep neural network based malware detection using two dimensional binary program features," in *2015 10th International Conference on Malicious and Unwanted Software (MALWARE)*, pp. 11–20, Fajardo, USA, 2015.

[11] M. Kalash, M. Rochan, N. Mohammed, N. D. B. Bruce, Y. Wang, and F. Iqbal, "Malware classification with deep convolutional neural networks," in *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5, Paris, France, 2018.

[12] X. Liu, J. Zhang, Y. Lin, and H. Li, "ATMPA: attacking machine learning-based malware visualization detection methods via adversarial examples," in *IEEE/ACM 27th International Symposium on Quality of Service (IWQoS)*, pp. 1–10, Phoenix Arizona, USA, 2019.

[13] H. Rathore, S. K. Sahay, P. Nikam, and M. Sewak, "Robust Android malware detection system against adversarial attacks using q-learning," *Information Systems Frontiers*, vol. 22, no. 8, pp. 1–20, 2020.

[14] C. Zhang, Y. Xu, Y. Hu, J. Wu, J. Ren, and Y. Zhang, "A blockchain-based multi-cloud storage data auditing scheme to locate faults," *IEEE Transactions on Cloud Computing*, 2021.

[15] D. Li, Q. Li, Y. Ye, and S. Xu, "A framework for enhancing deep neural networks against adversarial malware," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 1, pp. 736–750, 2021.

[16] W. Hu and Y. Tan, "Generating adversarial malware examples for black-box attacks based on GAN," 2017, http://arxiv.org/abs/1702.05983.

[17] H. S. Anderson, A. Kharkar, B. Filar, D. Evans, and P. Roth, "Learning to evade static PE machine learning malware models via reinforcement learning," 2018, http://arxiv.org/abs/1801.08917.

[18] K. Grosse, N. Papernot, P. Manoharan, M. Backes, and P. McDaniel, "Adversarial perturbations against deep neural networks for malware classification," 2016, http://arxiv.org/abs/1606.04435.

[19] O. Suciu, S. E. Coull, and J. Johns, "Exploring adversarial examples in malware detection," in *2019 IEEE Security and Privacy Workshops (SPW)*, pp. 8–14, San Francisco, CA, USA, 2019.

[20] B. Kolosnjaji, A. Demontis, B. Biggio et al., "Adversarial malware binaries: evading deep learning for malware detection in executables," in *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 533–537, Rome, Italy, 2018.

[21] M. Shafiq, Z. Tian, A. K. Bashir, X. du, and M. Guizani, "CorrAUC: a malicious bot-IoT traffic detection method in IoT network using machine-learning techniques," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3242–3254, 2021.

[22] S. Su, Z. Tian, S. Liang, S. Li, S. du, and N. Guizani, "A reputation management scheme for efficient malicious vehicle identification over 5G networks," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 46–52, 2020.

[23] C. Luo, Z. Tan, G. Min, J. Gan, W. Shi, and Z. Tian, "A novel web attack detection system for Internet of Things via ensemble classification," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5810–5818, 2021.

[24] M. Shafiq, Z. Tian, A. Bashir, A. Jolfaei, and X. Yu, "Data mining and machine learning methods for sustainable smart cities traffic classification: a survey," *Sustainable Cities and Society*, vol. 60, 2020.

[25] J. Yan, Y. Qi, and Q. Rao, "Detecting malware with an ensemble method based on deep neural network," *Security and Communication Networks*, vol. 2018, Article ID 7247095, 16 pages, 2018.

[26] X. Liu, Y. Lin, H. Li, and J. Zhang, "A novel method for malware detection on ML-based visualization technique," *Computers & Security*, vol. 89, 2020.

[27] W. Huang and J. W. Stokes, "MtNet: a multi-task neural network for dynamic malware classification," in *Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 399–418, Springer, 2016.

[28] B. Kolosnjaji, A. Zarras, G. Webster, and C. Eckert, "Deep learning for classification of malware system call sequences," in *Australasian Joint Conference on Artificial Intelligence*, pp. 137–149, Springer, 2016.

[29] R. Taheri, M. Ghahramani, R. Javidan, M. Shojafar, Z. Pooranian, and M. Conti, "Similarity-based Android malware detection using Hamming distance of static binary features," *Future Generation Computer Systems*, vol. 105, no. 4, pp. 230–247, 2020.

[30] H. Guo, S. Huang, C. Huang et al., "A lightweight cross-version binary code similarity detection based on similarity and correlation coefficient features," *IEEE Access*, vol. 8, pp. 120501–120512, 2020.

[31] N. A. Azeez, O. E. Odufuwa, S. Misra, J. Oluranti, and R. Damaševičius, "Windows PE malware detection using ensemble learning," *Informatics*, vol. 8, no. 1, pp. 10–20, 2021.

[32] E. Raff, J. Barker, J. Sylvester, R. Brandon, B. Catanzaro, and C. K. Nicholas, "Malware detection by eating a whole EXE," 2017, http://arxiv.org/abs/1710.09435.

[33] J. Lin, L. Xu, Y. Liu, and X. Zhang, "Black-box adversarial sample generation based on differential evolution," *Journal of Systems and Software*, vol. 170, no. 8, Article 110767, pp. 1–11, 2020.

[34] R. Taheri, R. Javidan, M. Shojafar, Z. Pooranian, A. Miri, and M. Conti, "On defending against label flipping attacks on malware detection systems," *Neural Computing and Applications*, vol. 32, no. 18, pp. 14781–14800, 2020.

[35] M. Li, Y. Sun, H. Lu, S. Maharjan, and Z. Tian, "Deep reinforcement learning for partially observable data poisoning

attack in crowdsensing systems," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6266–6278, 2020.

[36] R. Podschwadt and H. Takabi, "On effectiveness of adversarial examples and defenses for malware classification," in *International Conference on Security and Privacy in Communication Systems*, pp. 380–393, Springer, 2019.

[37] D. Maiorca, B. Biggio, and G. Giacinto, "Towards adversarial malware detection," *ACM Computing Surveys (CSUR)*, vol. 52, no. 4, pp. 1–36, 2019.

[38] F. Kreuk, A. Barak, S. Aviv-Reuven, M. Baruch, B. Pinkas, and J. Keshet, "Deceiving end-to-end deep learning malware detectors using adversarial examples," 2018, http://arxiv.org/abs/1802.04528.

[39] Z. Fang, J. Wang, B. Li, S. Wu, Y. Zhou, and H. Huang, "Evading anti-malware engines with deep reinforcement learning," *IEEE Access*, vol. 7, pp. 48867–48879, 2019.

[40] J. Yuan, S. Zhou, L. Lin, F. Wang, and J. Cui, "Black-box adversarial attacks against deep learning based malware binaries detection with GAN," in *the 24th European Conference on Artificial Intelligence (ECAI 2020)*, pp. 2536–2542, Santiago de Compostela, Spain, 2020.

[41] W. Song, X. Li, S. Afroz, D. Garg, D. Kuznetsov, and H. Yin, "Automatic generation of adversarial examples for interpreting malware classifiers," 2020, http://arxiv.org/abs/2003.03100.

[42] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2015, http://arxiv.org/abs/1412.6572.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8526206, 9 pages
https://doi.org/10.1155/2021/8526206

Research Article

LLSFIoT: Lightweight Logical Security Framework for Internet of Things

Isha Batra,[1] **Hatem S. A. Hamatta**,[2] **Arun Malik**,[1] **Mohammed Baz**,[3] **Fahad R. Albogamy**,[4] **Vishal Goyal**,[5] **and Sultan S. Alshamrani**[6]

[1] *School of Computer Science and Engineering, Lovely Professional University, Phagwara, India*
[2] *Department of Applied Sciences, Aqaba University College, Al Balqa Applied University, Aqaba, Jordan*
[3] *Department of Computer Engineering, College of Computer and Information Technology, Taif University, PO Box. 11099, Taif 21994, Saudi Arabia*
[4] *Turabah University College, Computer Sciences Program, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia*
[5] *Department of Computer Science, Punjabi University, Patiala, India*
[6] *Department of Information Technology, College of Computer and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia*

Correspondence should be addressed to Sultan S. Alshamrani; susamash@tu.edu.sa

Received 28 May 2021; Revised 17 July 2021; Accepted 30 July 2021; Published 18 August 2021

Academic Editor: Dr. Muhammad Shafiq

Current research in Internet of Things (IoT) is focused on the security enhancements to every communicated message in the network. Keeping this thought in mind, researcher in this work emphasizes on a security oriented cryptographic solution. Commonly used security cryptographic solutions are heavy in nature considering their key size, operations, and mechanism they follow to secure a message. This work first determines the benefit of applying lightweight security cryptographic solutions in IoT. The existing lightweight counterparts are still vulnerable to attacks and also consume calculative more power. Therefore, this research work proposes a new hybrid lightweight logical security framework for offering security in IoT (LLSFIoT). The operations, key size, and mechanism used in the proposed framework make its lightweight. The proposed framework is divided into three phases: registration, authentication, and light data security (LDS). LDS offers security by using unique keys at each round bearing small size. Key generation mechanism used is comparatively fast making the compromise of keys as a difficult task. These steps followed in the proposed algorithm design make it lightweight and a better solution for IoT-based networks as compared to the existing solutions that are relatively heavy weight in nature.

1. Introduction

A fresh primitive cryptography known as lightweight cryptography is specifically being put on the market for use as integrated systems in resource-restricted settings like radio frequency identification (RFID) IoT [1]. Lightweight will not be soft in nature, but will not be enforced on many apps. The attacker is restricted by lightweight algorithms with the exposure of only restricted information per key [2]. Lightweight alternatives are used to marinate the necessary trade between efficiency, safety, and assets [3]. The major challenges in IoT are restricted instruments such as RFID and battery-operated detectors. Particular consideration should therefore be paid to limiting the use of its funds and at the same moment to provide safety [4]. Solutions for lightweight cryptography deliver both safety and efficiency [5]. The easiest approach seems to be of IoT resource restrictions. Lightweight alternatives provide safety only through the exposure of restricted operational information [6]. The limitations in existing network are the use of large key size, block size, complex round structure, and the implementation requirements [7]. Being a resource constrained network, security in IoT should be using a security mechanism using less key size, block size, simple round structures, and simple implementation requirements [8, 9].

For Lightweight solutions, the National Institute of Standards and Technology (NIST) sets a minimum key size requirement of 112 bits. Even smaller key sizes are more vulnerable to brute force attack [10]. The following requirement is for a small block size. The lightweight cipher's block size should be smaller than that of conventional cyphers. For instance, if the block size is 64 bits rather than the 128 bits used by AES, a greater number of plaintext blocks can be encrypted [11]. Additionally, memory requirements will decrease. Following that, a simple round structure should be used: The rounds used in lightweight cyphers should be simpler than those used in conventional cryptography [12]. For instance, a round can be simplified by substituting a 4-bit S-Box for an 8-bit S-Box. This also reduces the amount of memory required. Increasing the total number of rounds to be fired may lower the amount of security that can be improvised [13]. The requirements take into account the fact that the device should be capable of either encryption or decryption. Rather than implementing the entire cypher, only required operations should be implemented [14, 15]. This comes out with an issue while implementing lightweight solutions in IoT but once it is implemented, the overall resources and life of network can be improved [16].

The contribution of this research work is to overcome the limitations of existing solutions by making changes in design of the security algorithm. In comparison to conventional block cyphers, the requirements of lightweight security solutions are lower for key size. Existing security solutions like AES, SIMON, and SPECK, they have more key size requirements as compared to the proposed algorithm.

The remaining paper is arranged accordingly. In Section II, work related to IoT security is reviewed. Existing solutions for providing security and authentication are covered. Section III propose the hybrid lightweight security solution for IoT comprising three phases, i.e, registration, authentication, and data security. Later, in Section IV, round key generation schedule of data security mechanism is discussed. Section V discusses the complete one round structure for Data security. Section VI analyzes the proposed algorithm by evaluating the mean and standard deviation. Finally, conclusion states the currents state of art and the benefit of the proposed security framework.

2. Related Work

Light weight means the algorithms that require fewer and optimal performance funds. The lightweight term does not refer to the weakness of the algorithm [17]. As the trend for future appliances with restricted systems has changed, a great deal of effort was produced to optimize AES for these apps. However, adaptation to the requirements of these systems in the AES was not suitable [18]. Although AES has been implemented quickly, however, it is still very complicated and has big codes that do not comply with the needs [19]. In [20], it is mentioned that AES is utilized as validation component in RFID-based frameworks. The AES is used in the application layer as an integrated COAP system. Advanced encryption standards (AES) is an institutionalized symmetrical square figure by NIST. It uses a replacement phase scheme and deals with a 128-bit square-length 4×4 network [21]. Each byte is influenced by the effects of subbytes, row shifting, MIXED COLUMNS, and ADD ROUND KEY. The key size that can be used is 128 bits, 192 bits, or 256 bits. AES is as yet defenseless against man-in-center assault [22, 23].

The author suggested in [24], PRESENT which is SPN based and used as an ultra-light safety calculation. It uses 4-bit info and S-box rates to advance devices at the replacement layer. It has 80 or 128 parts of main size and operates on 64 pieces. PRESENT is listed as a lightweight cryptography scheme in ISO/IEC 29192-2 : 2012 “Lightweight cryptography.” On 26 out of 31 rounds [25], PRESENT is indefensible from differential attack. In [26], author referred SIMON $2n$ an N-bit word cipher forming a $2n$-bit block. N can have 16, 24, 32, 48, and 64 values. SIMON $2n$ using key as k -word key (kn-bit) is referred as SIMON $2n$/kn. Therefore, SIMON 96/144 will be working on a block of 96-bit plaintext and using key of 144 bits. SIMON is a member of the block cypher family with varying block sizes. It can support 32, 48, 64, 96, and 128 bits of block size that further can work on varying key sizes.

In [27], author referred SPECK highlighted that SPECK requirements are like SIMON. SPECK 128/128 therefore means the 128 bit file length SPECK block code that sucks the 128-bit button. The SPECK supporting block and key size is identical to that of SIMON. SPECK uses Feistel structure performing bitwise XOR, circular shits, and modular addition in each round at both directions [28]. In [29], TWINE is described as a 64 bit block cipher forming a basic Feistel structure. Feistel functions consist of 16 4-bit subblocks using key addition. Two key sizes 80 and 128 bits are supported by TWINE. TWINE operates on total 36 rounds with same round function. In [30], author mentioned FANTOMAS as an LS-design example (LS consists of L-boxes using bit-sliced looking tables and S-boxes). The block cipher FANTOMAS can be displayed with the $s \times L$ bit array. The $s \times s$ parts are permutation for each matrix row, whereas the permutation for each matrix row is $L \times L$. Consider, for instance, a 128-bit FANTOMAS key and block length. The s-bits are 8, and the L-bits are 16.

3. Proposed Lightweight Logical Security Framework for IoT (LLSFIoT)

The proposed LLSFIoT is divided into three phases: registration, authentication, and LDS. When a new device enters the network, the credentials are first registered with the server using the key sharing mechanism. Once the device has the credentials, mutual authentication between the device and the server will take place before initiating any communication. Using the LDS algorithm, the data transmitted by and from the device is secured. The notations used in the process of registration, authenticationn and data security are shown in Table 1.

Table 1: Notations used in LLSFIoT.

Symbol	Description
D_i	ith device
IS	Information server
ID_D	Identity of device
ID_S	Identity of server
K_S	Key shared between device and server
SN	Sequence number
UID	Unique IDs
K_a	Alternate keys
T_V	Temporary variable
n	Number of bits in each word
$2n$	Block size/number of input bits
SK_i	ith key subblock
S^{-x}	Left rotation by x bits
S^y	Right rotation by y bits

3.1. Phase 1: Registration. Steps corresponding to registration phase are detailed below:

Step 1. Device D_i will initiate a connection that has been established with IS by submitting its IDD to the IS making use of a secure medium.

Step 2. IS following receipt of the connection request from D_i computes a nonce value N_S. This N_S is used to compute a shared key $K_{S,}$ where $K_S = \mathrm{ID}_S \oplus h(\mathrm{ID}_D \| N_S)$.

Step 3. Additionally to this, IS generates a collection of unique IDs, $\mathrm{UID} = \{\mathrm{id}_{1,}\ \mathrm{id}_{2,}\ \mathrm{id}_3, \cdots .\mathrm{id}_n.\}$, and set of alternate keys $Ka = \{k_{\mathrm{a}1}, k_{a2}, \cdots .k_{\mathrm{an}}\}$ in relation to one another $\mathrm{uid}_\mathrm{i} \in \mathrm{UID}$.

Step 4. Additionally, IS, a sequence number, is a generated randomly SN. As a result, for each request submitted by the D_i, IS generates K_S, unique IDs, alternate keys, and SN. If D_i makes an additional request to IS, a new SN is generated. The onus of IS is to maintain one copy of SN in database and forward same copy to the D_i. The benefit of using SN is to avoid any replay that the intruder may inject.

Step 5. Before the authentication process actually begins, IS checks to see if the SN sent by D_i matches one already stored in the database. Authentication phase 2 will be active when this match occurs. whereas IS ends connection with D_i and requires D_i to use one UID and Ka couple if match does not occur in SN.

The pair will be used once, and the entry will be removed in both the IS and D_i database. *I* will send a message at the end D_i encrypted using public key of D_i having a set of values: K_S, $\{\mathrm{id}_i, k_{\mathrm{ai}}\}$, SN, and in its own database keeps the same values as the D_i ID, i.e. IDD.

3.2. Phase 2: Authentication. In the authentication phase, two way mutual authentication is performed between D_i and IS. Steps corresponding to authentication phase are detailed below:

Step 6. D_i by taking a nonce value N_1 generates a variable $V_1 = h(\mathrm{ID}_D \| K_S \oplus N_1)$.

Step 2. Now, D_i creates a message of request having $\{V_1, \mathrm{ID}_D, \mathrm{SN}\}$ to the IS.

Step 3. On the off chance that SN is not accessible with D_i, D_i will use one of the $\{\mathrm{id}_i, \mathrm{k}_{\mathrm{ai}}\}$ pair where k_{ai} can be used in replacement of K_S.

Step 4. On receiving request from D_i, The IS verifies the message's SN or checks that additional parameters are legitimate or not if they match the matching SN of the D_i stored in the database. The value of N1 is later calculated by IS.

Step 5. If all the parameters are validated, then IS after taking a nonce value N_2 will generate a new random sequence number $\mathrm{SN}_{\mathrm{new}=}\ h\ (\mathrm{ID}_\mathrm{D} \| K_S \| N_1) \oplus \mathrm{SN}$ and computes a temporary variable $T_V = h(\mathrm{ID}_D \| K_S \| N_2) \oplus \mathrm{SN}_{\mathrm{new}}$ and computing variable $V\_2 = h(\mathrm{ID}_D \| K_S \| N_1 \| T_V)$.

Step 6. D_i on receiving message containing $\{V_2, \mathrm{SN}_{\mathrm{New}}, T_V\}$ from the IS computes the value $h(\mathrm{ID}_D \| K_S \| N_1 \| T_V)$ and compares it with V_2. If match occurs, D_i computes nonce N $_2$ using $T_V = h(\mathrm{ID}_D \| K_S \| N_2) \oplus \mathrm{SN}_{\mathrm{new.}}$

3.3. Phase 3: Lightweight Data Security (LDS) Algorithm. Once mutual authentication is performed between D_i and IS, the next step is to offer data security using the encryption method. Data is taken in blocks of 64 bits each, and the size of K_S shared between D_i and IS that is 128 bits. To offer security, a lightweight data security (LDS) algorithm is proposed. This algorithm takes of the secure data communication and offers the services for security such as confidentiality of data and integrity of data.

Proposed LDS works on 20 rounds using addition, rotation, and XOR (ARX) operations. This flexibility of choosing the number of rounds lies with the user depending upon the execution time required and also on full diffusion. The three operations ARX are chosen for offering optimum security trading off with lightweight solution considering the IoT application scenario. The reason for choosing only these operations for a round is discussed later in Section 4. The structure of LDS consisting of 20 rounds using ARX operations and a key generation function is represented through Figure 1.

4. Generation of Subkeys for Each Round

For each round, two n-bit subkey bocks are required, considering n as the number of bits in a word. Block size that can be taken as input will be $2n$. Here, block size of 64 bits is assumed; so, value of n is 32. Key size is taken as 128 bits. Therefore, for 20 rounds, 40 key subblocks have each of 32

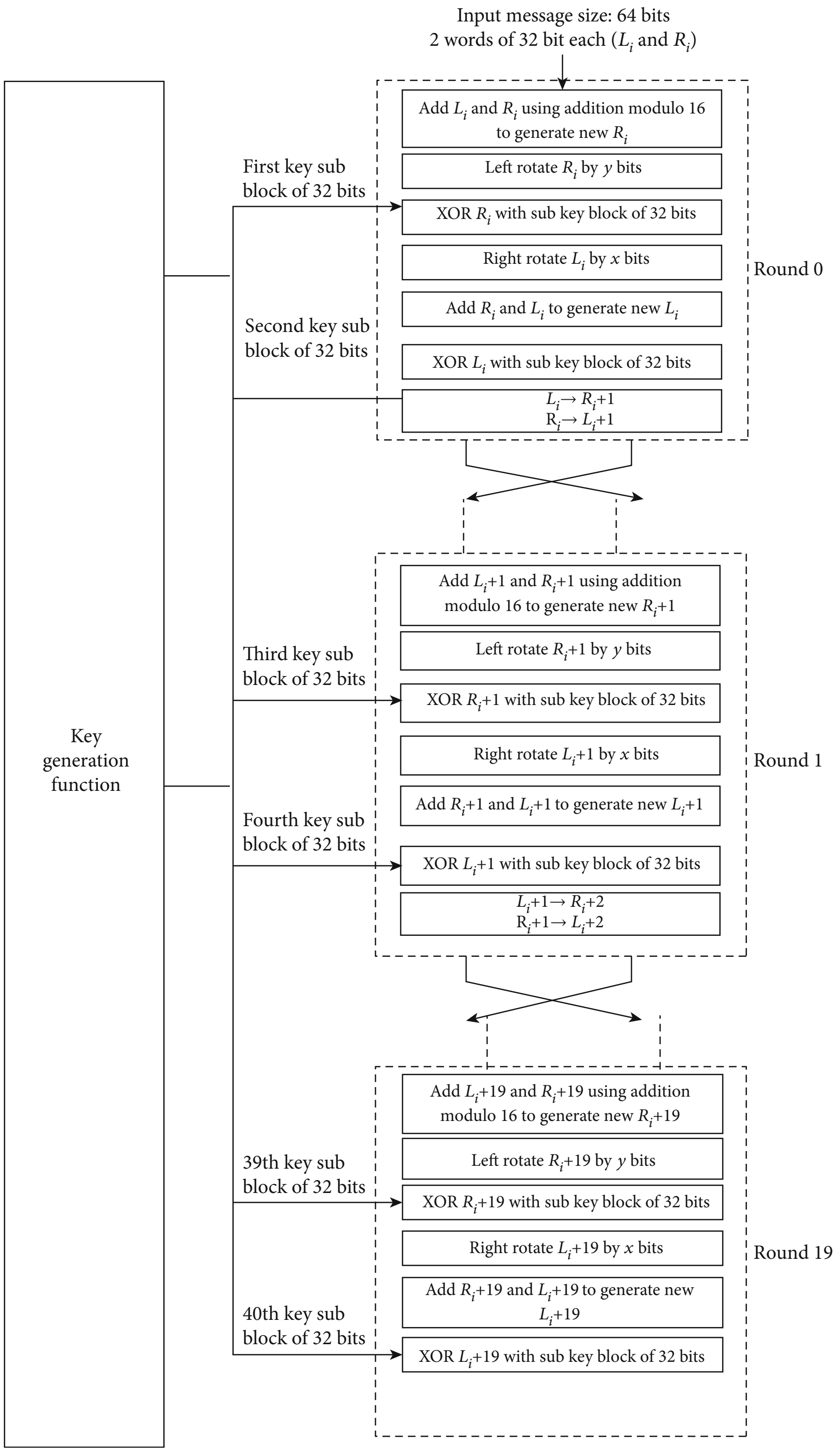

Input message size: 64 bits
2 words of 32 bit each (L_i and R_i)
Key generation function
First key sub block of 32 bits
Second key sub block of 32 bits
Add L_i and R_i using addition modulo 16 to generate new R_i
Left rotate R_i by y bits
XOR R_i with sub key block of 32 bits
Right rotate L_i by x bits
Add R_i and L_i to generate new L_i
XOR L_i with sub key block of 32 bits
$L_i \rightarrow R_i+1$
$R_i \rightarrow L_i+1$
Round 0
Third key sub block of 32 bits
Fourth key sub block of 32 bits
Add L_i+1 and R_i+1 using addition modulo 16 to generate new R_i+1
Left rotate R_i+1 by y bits
XOR R_i+1 with sub key block of 32 bits
Right rotate L_i+1 by x bits
Add R_i+1 and L_i+1 to generate new L_i+1
XOR L_i+1 with sub key block of 32 bits
$L_i+1 \rightarrow R_i+2$
$R_i+1 \rightarrow L_i+2$
Round 1
39th key sub block of 32 bits
40th key sub block of 32 bits
Add L_i+19 and R_i+19 using addition modulo 16 to generate new R_i+19
Left rotate R_i+19 by y bits
XOR R_i+19 with sub key block of 32 bits
Right rotate L_i+19 by x bits
Add R_i+19 and L_i+19 to generate new L_i+19
XOR L_i+19 with sub key block of 32 bits
Round 19

Figure 1: LDS structure.

bit out of 128 bit long key. A key generation mechanism is required for getting the key subblocks for each round of operation.

Subkey generation is done in such a manner that key generator gives a unique and random subkey every time it is run. For a good key generator mechanism, if the generated subkey is compromised by cryptanalysis, other subkeys should not be identified. The subkeys are generated from the main key of 128 bits. As stated earlier, each round requires two subkey blocks. The mechanism of key generation function consists of a key generation that divides the keys into subblocks. Key generator generates subkeys for two rounds at a time.

Therefore, for 20 rounds, key generator will work for 10 times and generate 4 subkey blocks each time, making a total of 40 subkey blocks. The whole mechanism of key generator is explained through following steps:

Step 1. The original key (K_i) of 128 bits is given as input to subkey generator.

Step 2. Sub key generator generates 4 sub key blocks of 32 bit each. Two key sub blocks of 32 bits are passed as input to first round and next two key sub blocks of 32 bits are passed as input to second round.

Step 3. Bits in original K_i are processed using a mixing function to generate input for the running the key generator for the next time. From there again, 4 key subblocks are generated for next two rounds.

Step 4. Mixing function takes as input the output of the previous key generator function. For the first time, after the execution of key generator, original K_i consists of 4 key subblocks, let us say, SK_1, SK_2, SK_3, SK_4, each of 32 bits. Mixing function performs the XOR operation in circular rotation. All the bits of SK_1 are XORed with random bits of SK_2, SK_2 is XORed with random bits of SK_3, SK_3 is XORed with random bits of $\mathrm{SK}_{4,}$ and SK_4 is XORed with random bits of SK_1. $\mathrm{SK}_1(0)$ represents the first bit of key subblock SK_1. Figure 2 shows the block diagram for operation of key generator. The sample equations to generate subkey can be represented mathematically in Table 2.

Step 5. Step 3 and step 4 are repeated till all the 40 subkey blocks are generated for all the 20 rounds.

5. Round Function of LDS

LDS framework works on the Feistel-like structure. Operations used during the encryption process of LDS are

(i) Addition modulo 2^n, considering n as the number of bits in a word. If n is 16, block size will be 32 and for n taken as 32, block size will be 64 bits. Addition modulo is preferred over multiplication modulo. There may be multiple reasons for choosing addition over multiplication. First, multiplication require more cycles as compared to addition even with the fastest CPUs. Second, operation of multiplication may lead to timing attacks

(ii) Bitwise XOR, ⊕: most block ciphers work using XOR as the basic operation as compared to other operations like AND and OR. Numbers of factors supporting XOR over other operations are first, XOR operation works on reversible procedure. When encryption is performed on original text XOR with key to generate cipher text, same key when operated using XOR with cipher text the resultant will be same original text. Second, XOR can be realized using the NAND gate requiring few transistors as compared to other operations, making its hardware implementation quite easier. Third, in XOR, the output is dependent on both the operands as compared to AND and OR. In AND, if one of the operand is false, second is not evaluated at all. In OR, if one of the operand is true and second is not evaluated at all, whereas, in XOR, if first operand is true or false, second needs to be evaluated for getting the expected output

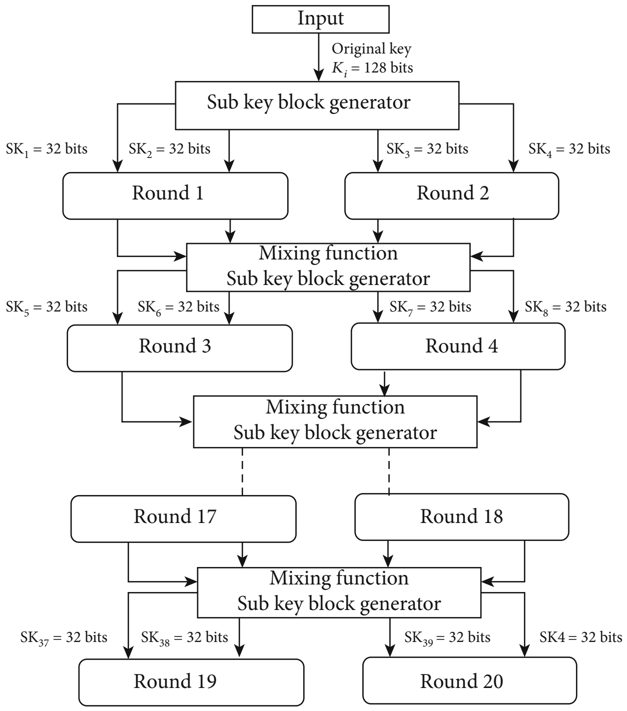


Figure 2: Block diagram for key subblock generator function.

(iii) R^{-b} and R^b are left and right rotations respectively, where b is the number of bits to rotate. Rotations are preferred over shift as rotation when used with the XOR operation that creates maximum diffusion in the resultant output with alteration in a single input bit. On the other hand, when shift is used with the XOR, then diffusion created is less in output with alteration in a single input bit

The input block of n bits is divided into two equal halves. For example, if input text is 64 bits long, it will be divided into 32 bits each represented as L_i and R_i. L_i represents the left subblock, and R_i represents the right subblock. The left and the right subblock in a particular round is evaluated as

$$\begin{aligned} L_i &= (S^{-x}L_i + (S^y(L_i + R_i) \oplus SK_2)) \oplus SK_1, \\ R_i &= S^y(L_i + R_i) \oplus SK_2. \end{aligned} \tag{1}$$

Therefore, the round function of LDS is denoted as

$$F(L_i, R_i) = ((S^{-x}Li + (S^y(L_i + R_i) \oplus \mathrm{SK}_2)) \oplus \mathrm{SK}_1, S^y(L_i + R_i) \oplus \mathrm{SK}_2), \tag{2}$$

Table 2

$\mathrm{SK}_1(0) = \mathrm{SK}_1(0) \oplus \mathrm{SK}_2(31)$	$\mathrm{SK}_2(0) = \mathrm{SK}_2(0) \oplus \mathrm{SK}_3(0)$
$\mathrm{SK}_1(1) = \mathrm{SK}_1(1) \oplus \mathrm{SK}_2(0)$	$\mathrm{SK}_2(1) = \mathrm{SK}_2(1) \oplus \mathrm{SK}_3(1)$
$\mathrm{SK}_1(2) = \mathrm{SK}_1(2) \oplus \mathrm{SK}_2(1)$	$\mathrm{SK}_2(2) = \mathrm{SK}_2(2) \oplus \mathrm{SK}_3(2)$
$\mathrm{SK}_1(3) = \mathrm{SK}_1(3) \oplus \mathrm{SK}_2(2)$	$\mathrm{SK}_2(3) = \mathrm{SK}_2(3) \oplus \mathrm{SK}_3(3)$
$\mathrm{SK}_1(4) = \mathrm{SK}_1(4) \oplus \mathrm{SK}_2(3)$	$\mathrm{SK}_2(4) = \mathrm{SK}_2(4) \oplus \mathrm{SK}_3(4)$
$\mathrm{SK}_1(5) = \mathrm{SK}_1(5) \oplus \mathrm{SK}_2(4)$	$\mathrm{SK}_2(5) = \mathrm{SK}_2(5) \oplus \mathrm{SK}_3(5)$
$\mathrm{SK}_1(6) = \mathrm{SK}_1(6) \oplus \mathrm{SK}_2(5)$	$\mathrm{SK}_2(6) = \mathrm{SK}_2(6) \oplus \mathrm{SK}_3(6)$
$\mathrm{SK}_1(7) = \mathrm{SK}_1(7) \oplus \mathrm{SK}_2(6)$	$\mathrm{SK}_2(7) = \mathrm{SK}_2(7) \oplus \mathrm{SK}_3(7)$
$\mathrm{SK}_1(8) = \mathrm{SK}_1(8) \oplus \mathrm{SK}_2(7)$	$\mathrm{SK}_2(8) = \mathrm{SK}_2(8) \oplus \mathrm{SK}_3(8)$
$\mathrm{SK}_1(9) = \mathrm{SK}_1(9) \oplus \mathrm{SK}_2(8)$	$\mathrm{SK}_2(9) = \mathrm{SK}_2(9) \oplus \mathrm{SK}_3(9)$
$\mathrm{SK}_1(10) = \mathrm{SK}_1(10) \oplus \mathrm{SK}_2(9)$	$\mathrm{SK}_2(10) = \mathrm{SK}_2(10) \oplus \mathrm{SK}_3(10)$
$\mathrm{SK}_1(11) = \mathrm{SK}_1(11) \oplus \mathrm{SK}_2(10)$	$\mathrm{SK}_2(11) = \mathrm{SK}_2(11) \oplus \mathrm{SK}_3(11)$
$\mathrm{SK}_1(12) = \mathrm{SK}_1(12) \oplus \mathrm{SK}_2(11)$	$\mathrm{SK}_2(12) = \mathrm{SK}_2(12) \oplus \mathrm{SK}_3(12)$
$\mathrm{SK}_1(13) = \mathrm{SK}_1(13) \oplus \mathrm{SK}_2(12)$	$\mathrm{SK}_2(13) = \mathrm{SK}_2(13) \oplus \mathrm{SK}_3(13)$
$\mathrm{SK}_1(14) = \mathrm{SK}_1(14) \oplus \mathrm{SK}_2(13)$	$\mathrm{SK}_2(14) = \mathrm{SK}_2(14) \oplus \mathrm{SK}_3(14)$
$\mathrm{SK}_1(15) = \mathrm{SK}_1(15) \oplus \mathrm{SK}_2(14)$	$\mathrm{SK}_2(15) = \mathrm{SK}_2(15) \oplus \mathrm{SK}_3(15)$
$\mathrm{SK}_1(16) = \mathrm{SK}_1(16) \oplus \mathrm{SK}_2(15)$	$\mathrm{SK}_2(16) = \mathrm{SK}_2(16) \oplus \mathrm{SK}_3(16)$
$\mathrm{SK}_1(17) = \mathrm{SK}_1(17) \oplus \mathrm{SK}_2(16)$	$\mathrm{SK}_2(17) = \mathrm{SK}_2(17) \oplus \mathrm{SK}_3(17)$
$\mathrm{SK}_1(18) = \mathrm{SK}_1(18) \oplus \mathrm{SK}_2(17)$	$\mathrm{SK}_2(18) = \mathrm{SK}_2(18) \oplus \mathrm{SK}_3(18)$
$\mathrm{SK}_1(19) = \mathrm{SK}_1(19) \oplus \mathrm{SK}_2(18)$	$\mathrm{SK}_2(19) = \mathrm{SK}_2(19) \oplus \mathrm{SK}_3(19)$
$\mathrm{SK}_1(20) = \mathrm{SK}_1(20) \oplus \mathrm{SK}_2(19)$	$\mathrm{SK}_2(20) = \mathrm{SK}_2(20) \oplus \mathrm{SK}_3(20)$
$\mathrm{SK}_1(21) = \mathrm{SK}_1(21) \oplus \mathrm{SK}_2(20)$	$\mathrm{SK}_2(21) = \mathrm{SK}_2(21) \oplus \mathrm{SK}_3(21)$
$\mathrm{SK}_1(22) = \mathrm{SK}_1(22) \oplus \mathrm{SK}_2(21)$	$\mathrm{SK}_2(22) = \mathrm{SK}_2(22) \oplus \mathrm{SK}_3(22)$
$\mathrm{SK}_1(23) = \mathrm{SK}_1(23) \oplus \mathrm{SK}_2(22)$	$\mathrm{SK}_2(23) = \mathrm{SK}_2(23) \oplus \mathrm{SK}_3(23)$
$\mathrm{SK}_1(24) = \mathrm{SK}_1(24) \oplus \mathrm{SK}_2(23)$	$\mathrm{SK}_2(24) = \mathrm{SK}_2(24) \oplus \mathrm{SK}_3(24)$
$\mathrm{SK}_1(25) = \mathrm{SK}_1(25) \oplus \mathrm{SK}_2(24)$	$\mathrm{SK}_2(25) = \mathrm{SK}_2(25) \oplus \mathrm{SK}_3(25)$
$\mathrm{SK}_1(26) = \mathrm{SK}_1(26) \oplus \mathrm{SK}_2(25)$	$\mathrm{SK}_2(26) = \mathrm{SK}_2(26) \oplus \mathrm{SK}_3(26)$
$\mathrm{SK}_1(27) = \mathrm{SK}_1(27) \oplus \mathrm{SK}_2(26)$	$\mathrm{SK}_2(27) = \mathrm{SK}_2(27) \oplus \mathrm{SK}_3(27)$
$\mathrm{SK}_1(28) = \mathrm{SK}_1(28) \oplus \mathrm{SK}_2(27)$	$\mathrm{SK}_2(28) = \mathrm{SK}_2(28) \oplus \mathrm{SK}_3(28)$
$\mathrm{SK}_1(29) = \mathrm{SK}_1(29) \oplus \mathrm{SK}_2(28)$	$\mathrm{SK}_2(29) = \mathrm{SK}_2(29) \oplus \mathrm{SK}_3(29)$
$\mathrm{SK}_1(30) = \mathrm{SK}_1(30) \oplus \mathrm{SK}_2(29)$	$\mathrm{SK}_2(30) = \mathrm{SK}_2(30) \oplus \mathrm{SK}_3(30)$
$\mathrm{SK}_1(31) = \mathrm{SK}_1(31) \oplus \mathrm{SK}_2(30)$	$\mathrm{SK}_2(31) = \mathrm{SK}_2(31) \oplus \mathrm{SK}_3(31)$
$\mathrm{SK}_3(0) = \mathrm{SK}_3(0) \oplus \mathrm{SK}_4(1)$	$\mathrm{SK}_4(0) = \mathrm{SK}_4(0) \oplus \mathrm{SK}_1(2)$
$\mathrm{SK}_3(1) = \mathrm{SK}_3(1) \oplus \mathrm{SK}_4(2)$	$\mathrm{SK}_4(1) = \mathrm{SK}_4(1) \oplus \mathrm{SK}_1(3)$
$\mathrm{SK}_3(2) = \mathrm{SK}_3(2) \oplus \mathrm{SK}_4(3)$	$\mathrm{SK}_4(2) = \mathrm{SK}_4(2) \oplus \mathrm{SK}_1(4)$
$\mathrm{SK}_3(3) = \mathrm{SK}_3(3) \oplus \mathrm{SK}_4(4)$	$\mathrm{SK}_4(3) = \mathrm{SK}_4(3) \oplus \mathrm{SK}_1(5)$
$\mathrm{SK}_3(4) = \mathrm{SK}_3(4) \oplus \mathrm{SK}_4(5)$	$\mathrm{SK}_4(4) = \mathrm{SK}_4(4) \oplus \mathrm{SK}_1(6)$
$\mathrm{SK}_3(5) = \mathrm{SK}_3(5) \oplus \mathrm{SK}_4(6)$	$\mathrm{SK}_4(5) = \mathrm{SK}_4(5) \oplus \mathrm{SK}_1(7)$
$\mathrm{SK}_3(6) = \mathrm{SK}_3(6) \oplus \mathrm{SK}_4(7)$	$\mathrm{SK}_4(6) = \mathrm{SK}_4(6) \oplus \mathrm{SK}_1(8)$
$\mathrm{SK}_3(7) = \mathrm{SK}_3(7) \oplus \mathrm{SK}_4(8)$	$\mathrm{SK}_4(7) = \mathrm{SK}_4(7) \oplus \mathrm{SK}_1(9)$
$\mathrm{SK}_3(8) = \mathrm{SK}_3(8) \oplus \mathrm{SK}_4(9)$	$\mathrm{SK}_4(8) = \mathrm{SK}_4(8) \oplus \mathrm{SK}_1(10)$
$\mathrm{SK}_3(9) = \mathrm{SK}_3(9) \oplus \mathrm{SK}_4(10)$	$\mathrm{SK}_4(9) = \mathrm{SK}_4(9) \oplus \mathrm{SK}_1(11)$
$\mathrm{SK}_3(10) = \mathrm{SK}_3(10) \oplus \mathrm{SK}_4(11)$	$\mathrm{SK}_4(10) = \mathrm{SK}_4(10) \oplus \mathrm{SK}_1(12)$
$\mathrm{SK}_3(11) = \mathrm{SK}_3(11) \oplus \mathrm{SK}_4(12)$	$\mathrm{SK}_4(11) = \mathrm{SK}_4(11) \oplus \mathrm{SK}_1(13)$
$\mathrm{SK}_3(12) = \mathrm{SK}_3(12) \oplus \mathrm{SK}_4(13)$	$\mathrm{SK}_4(12) = \mathrm{SK}_4(12) \oplus \mathrm{SK}_1(14)$
$\mathrm{SK}_3(13) = \mathrm{SK}_3(13) \oplus \mathrm{SK}_4(14)$	$\mathrm{SK}_4(13) = \mathrm{SK}_4(13) \oplus \mathrm{SK}_1(15)$
$\mathrm{SK}_3(14) = \mathrm{SK}_3(14) \oplus \mathrm{SK}_4(15)$	$\mathrm{SK}_4(14) = \mathrm{SK}_4(14) \oplus \mathrm{SK}_1(16)$
$\mathrm{SK}_3(15) = \mathrm{SK}_3(15) \oplus \mathrm{SK}_4(16)$	$\mathrm{SK}_4(15) = \mathrm{SK}_4(15) \oplus \mathrm{SK}_1(17)$
$\mathrm{SK}_3(16) = \mathrm{SK}_3(16) \oplus \mathrm{SK}_4(17)$	$\mathrm{SK}_4(16) = \mathrm{SK}_4(16) \oplus \mathrm{SK}_1(18)$
$\mathrm{SK}_3(17) = \mathrm{SK}_3(17) \oplus \mathrm{SK}_4(18)$	$\mathrm{SK}_4(17) = \mathrm{SK}_4(17) \oplus \mathrm{SK}_1(19)$
$\mathrm{SK}_3(18) = \mathrm{SK}_3(18) \oplus \mathrm{SK}_4(19)$	$\mathrm{SK}_4(18) = \mathrm{SK}_4(18) \oplus \mathrm{SK}_1(20)$
$\mathrm{SK}_3(19) = \mathrm{SK}_3(19) \oplus \mathrm{SK}_4(20)$	$\mathrm{SK}_4(19) = \mathrm{SK}_4(19) \oplus \mathrm{SK}_1(21)$
$\mathrm{SK}_3(20) = \mathrm{SK}_3(20) \oplus \mathrm{SK}_4(21)$	$\mathrm{SK}_4(20) = \mathrm{SK}_4(20) \oplus \mathrm{SK}_1(22)$
$\mathrm{SK}_3(21) = \mathrm{SK}_3(21) \oplus \mathrm{SK}_4(22)$	$\mathrm{SK}_4(21) = \mathrm{SK}_4(21) \oplus \mathrm{SK}_1(23)$
$\mathrm{SK}_3(22) = \mathrm{SK}_3(22) \oplus \mathrm{SK}_4(23)$	$\mathrm{SK}_4(22) = \mathrm{SK}_4(22) \oplus \mathrm{SK}_1(24)$
$\mathrm{SK}_3(23) = \mathrm{SK}_3(23) \oplus \mathrm{SK}_4(24)$	$\mathrm{SK}_4(23) = \mathrm{SK}_4(23) \oplus \mathrm{SK}_1(25)$
$\mathrm{SK}_3(24) = \mathrm{SK}_3(24) \oplus \mathrm{SK}_4(25)$	$\mathrm{SK}_4(24) = \mathrm{SK}_4(24) \oplus \mathrm{SK}_1(26)$
$\mathrm{SK}_3(25) = \mathrm{SK}_3(25) \oplus \mathrm{SK}_4(26)$	$\mathrm{SK}_4(25) = \mathrm{SK}_4(25) \oplus \mathrm{SK}_1(27)$
$\mathrm{SK}_3(26) = \mathrm{SK}_3(26) \oplus \mathrm{SK}_4(27)$	$\mathrm{SK}_4(26) = \mathrm{SK}_4(26) \oplus \mathrm{SK}_1(28)$
$\mathrm{SK}_3(27) = \mathrm{SK}_3(27) \oplus \mathrm{SK}_4(28)$	$\mathrm{SK}_4(27) = \mathrm{SK}_4(27) \oplus \mathrm{SK}_1(29)$
$\mathrm{SK}_3(28) = \mathrm{SK}_3(28) \oplus \mathrm{SK}_4(29)$	$\mathrm{SK}_4(28) = \mathrm{SK}_4(28) \oplus \mathrm{SK}_1(30)$
$\mathrm{SK}_3(29) = \mathrm{SK}_3(29) \oplus \mathrm{SK}_4(30)$	$\mathrm{SK}_4(29) = \mathrm{SK}_4(29) \oplus \mathrm{SK}_1(31)$
$\mathrm{SK}_3(30) = \mathrm{SK}_3(30) \oplus \mathrm{SK}_4(31)$	$\mathrm{SK}_4(30) = \mathrm{SK}_4(30) \oplus \mathrm{SK}_1(0)$
$\mathrm{SK}_3(31) = \mathrm{SK}_3(31) \oplus \mathrm{SK}_4(0)$	$\mathrm{SK}_4(31) = \mathrm{SK}_4(31) \oplus \mathrm{SK}_1(1)$

where x and y are the rotation constants. For block size of 64 bits and key size of 128 bits, the value of x is taken as 7, and the value of y is taken as 3. This composition of round function is represented through Figure 3.

6. Evaluating Diffusion Property for LDS Round

Diffusion property of the cryptographic algorithm is focused on incorporating the avalanche effect. It refers to observe the change in number of bits of the output cipher text with a single bit modification in the input original text. With more number of bits affected by diffusion, the cryptographic solution proves to be stronger.

The input original text of 64 bits is divided into 2 subdata blocks of 32 bits each and referred as A and B. The values of rotation constant for creating the diffusion matrix may vary from 0 to 31. Therefore, the total possible combinations of rotation constants x and y each carrying value from 0 to 31 may lead to $32 \times 32 = 1024$ combinations. 1024 combinations can generate 1024 diffusion matrices based on respective designs. A sample diffusion matrix is shown in Table 3 below:

In Table 3, M_{AA} refers to the count of number of bits modified in A by modifying the single bit of A. As there are 32 bits in A, the mean value and the standard deviation are calculated after changing every bit of A and noticing its effect on A and B. Similar effect can be noticed in M_{AB}, M_{BA}, and M_{BB}.

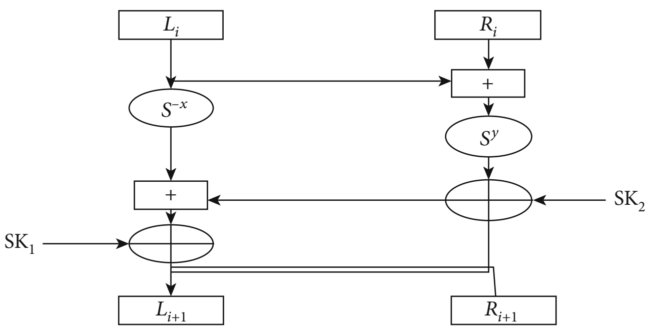


Figure 3: Single round function of LDS.

Table 3: Generalized diffusion table.

Input	Output	
	Left block (A)	Right block (B)
Left block (A)	M_{AA}	M_{AB}
Right block (B)	M_{BA}	M_{BB}

Table 4: Diffusion table considering $x = 7$ and $y = 3$.

Input	Output	
	Left block (A)	Right block (B)
Left block (A)	10.5	12.3125
Right block (B)	6.71815	8.90625
Mean = 9.609, standard deviation = 2.058		

In order to generate the diffusion matrix, certain steps are taken that require the input and rotation constants assumed as x and y here.

Step 1. Input block of 64 bits is divided into two subblocks referred as A and B each of 32 bits.

Step 2. Considering the different combinations of x and y where each can take values from 0 to 31, thus, the overall possible combinations are 1024. Here, the value of x and y is assumed to be fixed; that is, x is taken as 7, and y is taken as 3, while executing the LDS algorithm.

Step 3. Round function of LDS is executed over the data blocks A and B to generate output as A^1 and B^1.

Step 4. Start by modifying one bit of A, then execute LDS over modified A bits and the original B to get output as A^2 and B^2. Compare bits of A^1 with bits of A^2 and calculate the number of bits which have been altered, that will become value of M_{AA}. Similarly, compare bits of B^1 with bits of B^2 and calculate the number of bits which have been altered, that will become value of M_{AB}.

Step 5. Repeat step 3 but this time by modifying one bit of B. Execute LDS over modified B bits and the original A to get output as A^2 and B^2. Compare bits of A^1 with bits of A^2 and calculate the number of bits which have been altered, that will become the value of M_{BA}. Similarly, compare bits of B^1 with bits of B^2 and calculate the number of bits which have been altered, that will become the value of M_{BB}.

Table 5: Diffusion table considering $x = 8$ and $y = 3$.

Input	Output	
	Left block (A)	Right block (B)
Left block (A)	9.375	10.156
Right block (B)	3.156	7.218
Mean = 7.4762, standard deviation = 2.716		

Table 6: Diffusion table considering $x = 7$ and $y = 2$.

Input	Output	
	Left block (A)	Right block (B)
Left block (A)	8.625	11.25
Right block (B)	4.312	6.75
Mean = 7.734, standard deviation = 2.541		

Table 7: Mean and standard deviation with different sets of rotation constants.

Rotation constants	Mean	Standard deviation
$x = 7$ $y = 3$	9.609	2.058
$x = 8$ $y = 3$	7.476	2.716
$x = 7$ $y = 2$	7.734	2.541

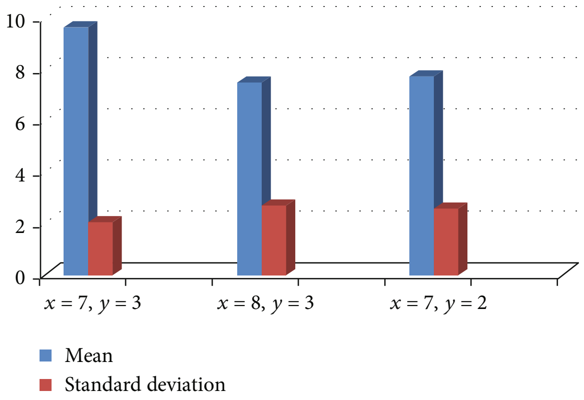


Figure 4: Mean and standard deviation for different sets of rotation constants.

Step 6. Repeat steps 3 and 4 at least 64 times using rotation constant x =7 and $y = 3$ to find the average of each value of matrix M to get diffusion table as shown in Table 4 below.

The possible combinations for x and y can be 1024, but the same steps for creating diffusion table are repeated by considering the most common used rotation constants $x = 8$ and $y = 3$ and second time by considering $x = 7$ and $y = 2$. The diffusion table generated by repeating the steps 64 times for the rotation constants $x = 8$ and $y = 3$ is shown through Table 5 below.

The diffusion table generated by repeating the steps 64 times for the rotation constants $x = 7$ and $y = 2$ is shown through Table 6.

Mean value in all the combinations shows the average number of bits that are affected by changing the individual bits as shown in equation (3)

$$\text{Mean} = (M\text{AA} + \text{MAB} + \text{MBB} + \text{MBA})/4. \tag{3}$$

Standard deviation is calculated by finding the variance after subtracting each data value from the mean and then finding their sum and finally performing square root as shown in equation (4).

$$\text{Standard Deviation} = \sqrt{\sum_{i,j=1}^{2} (\text{Mij} - \text{Mean})}. \tag{4}$$

Rotation constants chosen for creating diffusion matrix and hence calculating mean and standard deviation show the extent of output change by changing input bits. These constants are used to calculate transition probability. For block size of 64 bit block and key size of 128 bits, the mean and standard deviation for three sets of rotation constants are shown in Table 7 and are represented through Figure 4.

Figure 4 clarifies that when same LDS is executed with different combinations of rotation constants, the maximum value and the least standard deviation are observed with rotation constants $x = 7$ and $y = 3$. Therefore, the proposed LDS algorithm chooses the rotation constants $x = 7$ and $y = 3$.

7. Conclusion

Once the data is collected through devices using sensors, the next concern is to offer security to data or devices that play active role in communication. Therefore, this research work proposes a LLSFIoT model consisting of three phases. Phase 1 registers the new devices with the central server and handover the essential credentials to the device. Phase 2 performs mutual authentication between server and the device. Phase 3 proposed a LDS algorithm that offers confidentiality and integrity to data in transit. LDS works as a Feistel structure on 20 rounds of operation using ARX operations: addition, rotation, and XOR. The LDS is evaluated by performing cryptanalysis using diffusion property. Different sets of rotation constants are used to find mean and standard deviation. This research work concludes that LDS works well with constant value as $x = 7$ and $y = 3$ with maximum mean and minimum standard deviation assuring that single change in input will affect more bits in output.

Data Availability

There is no dataset involved in this research paper.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This research was supported by the Taif University Research Supporting Project number (TURSP-2020/239), Taif University, Taif, Saudi Arabia.

References

[1] C. Cheng, R. Lu, A. Petzoldt, and T. Takagi, “Securing the Internet of Things in a quantum world,” *IEEE Communications Magazine*, vol. 55, no. 2, pp. 116–120, 2017.

[2] W. Feng, Y. Qin, S. Zhao, and D. Feng, “AAoT: Lightweight attestation and authentication of low-resource things in IoT and CPS,” *Computer Networks*, vol. 134, pp. 167–182, 2018.

[3] J. S. Silva, P. Zhang, T. Pering, F. Boavida, T. Hara, and N. C. Liebau, “People-Centric Internet of Things,” *IEEE Communications Magazine*, vol. 55, no. 2, pp. 18-19, 2017.

[4] C. Verikoukis, R. Minerva, M. Guizani, S. K. Datta, Y. K. Chen, and H. A. Muller, “Internet of Things: part 2,” *IEEE Communications Magazine*, vol. 55, no. 2, pp. 114-115, 2017.

[5] K. R. Choo, S. Gritzalis, and J. H. Park, “Cryptographic solutions for industrial internet-of-things: research challenges and opportunities,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3567–3569, 2018.

[6] L. Ledwaba, G. P. Hancke, H. S. Venter, and S. J. Isaac, “Performance costs of software cryptography in securing new-generation internet of energy endpoint devices,” *IEEE Access*, vol. 6, pp. 9303–9323, 2018.

[7] Y. Harbi, Z. Aliouat, S. Harous, A. Bentaleb, and A. Refoufi, “A review of security in Internet of Things,” *Wireless Personal Communications*, vol. 108, no. 1, pp. 325–344, 2019.

[8] Hong, “Internet of Things is now in sync with real life,” 2016, https://news. http://samsung.com/global/samsung-shows-that-the-internet-of-things-is-now-in-sync-with-real-life.

[9] W. H. Hassan, “Current research on Internet of Things (IoT) security: a survey,” *Computer Networks*, vol. 148, pp. 283–294, 2019.

[10] S. S. Dhanda, B. Singh, and P. Jindal, “Lightweight cryptography: A solution to secure IoT,” *Wireless Personal Communications*, vol. 112, no. 3, pp. 1947–1980, 2020.

[11] Y. Yang, H. Peng, L. Li, and X. Niu, “General theory of security and a study case in internet of things,” *IEEE Internet of Things Journal*, vol. 4, no. 2, pp. 592–600, 2017.

[12] P. I. Radoglou Grammatikis, P. G. Sarigiannidis, and I. D. Moscholios, “Securing the Internet of Things: challenges, threats and solutions,” *Internet of Things*, vol. 5, pp. 41–70, 2019.

[13] N. Miloslavskaya and A. Tolstoy, “Internet of things: information security challenges and solutions,” *Cluster Computing*, vol. 22, no. 1, pp. 103–119, 2019.

[14] V. Adat and B. B. Gupta, “Security in Internet of Things: issues, challenges, taxonomy, and architecture,” *Telecommunication Systems*, vol. 67, no. 3, pp. 423–441, 2018.

[15] A. Jan, S. A. Parah, B. A. Malik, and M. Rashid, “Secure data transmission in IoTs based on CLoG edge detection,” *Future Generation Computer Systems*, vol. 121, pp. 59–73, 2021.

[16] M. Rashid and U. I. Wani, “Role of fog computing platform in analytics of Internet of Things- issues, challenges and opportunities,” *Fog, Edge, and Pervasive Computing in Intelligent IoT Driven Applications*, vol. 1, pp. 209–220, 2020.

[17] R. Morabito, V. Cozzolino, A. Y. Ding, N. Beijar, and J. Ott, “Consolidate IoT edge computing with lightweight virtualization,” *IEEE Network*, vol. 32, no. 1, pp. 102–111, 2018.

[18] H. Hellaoui, M. Koudil, and A. Bouabdallah, “Energy-efficient mechanisms in security of the internet of things: a survey,” *Computer Networks*, vol. 127, pp. 173–189, 2017.

[19] D. He, R. Ye, S. Chan, M. Guizani, and Y. Xu, “Privacy in the Internet of Things for smart healthcare,” *IEEE Communications Magazine*, vol. 56, no. 4, pp. 38–44, 2018.

[20] N. Karthik and V. S. Ananthanarayana, “Trust based data gathering in wireless sensor network,” *Wireless Personal Communications*, vol. 108, no. 3, pp. 1697–1717, 2019.

[21] P. Derbez and P. A. Fouque, “Exhausting Demirci-Selçuk meet-in-the-middle attacks against reduced-round AES,” in *International Workshop on Fast Software Encryption. FSE 2013*, S. Moriai, Ed., vol. 8424 of Lecture Notes in Computer Science, pp. 541–560, Springer, Berlin, Heidelberg, 2014.

[22] W. Li, H. Song, and F. Zeng, “Policy-based secure and trustworthy sensing for internet of things in smart cities,” *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 716–723, 2018.

[23] Y. Kawamoto, H. Nishiyama, N. Kato, Y. Shimizu, A. Takahara, and T. Jiang, “Effectively collecting data for the location-based authentication in internet of things,” *IEEE Systems Journal*, vol. 11, no. 3, pp. 1403–1411, 2017.

[24] Z. Ling, J. Luo, Y. Xu, C. Gao, K. Wu, and X. Fu, “Security vulnerabilities of internet of things: a case study of the smart plug system,” *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 1899–1909, 2017.

[25] C. Blondeau and K. Nyberg, “Links Between truncated differential and multidimensional linear properties of block ciphers and underlying attack complexities,” in *Advances in Cryptology – EUROCRYPT 2014. EUROCRYPT 2014*, P. Q. Nguyen and E. Oswald, Eds., vol. 8441 of Lecture Notes in Computer Science, pp. 165–182, Springer, Berlin, Heidelberg, 2014.

[26] R. Beaulieu, S. T. Clark, D. Shors, B. Weeks, J. Smith, and L. Wingers, “The SIMON and SPECK lightweight block ciphers,” in *Proceedings of the 52nd Annual Design Automation Conference*, pp. 1–6, San Francisco, CA, USA, 2015.

[27] A. V. Duka and B. Genge, “Implementation of SIMON and SPECK lightweight block ciphers on programmable logic controllers,” in *2017 5th International Symposium on Digital Forensic and Security (ISDFS)*, pp. 1–6, Tirgu Mures, Romania, 2017.

[28] N. Wang, X. Wang, K. Jia, and J. Zhao, “Differential attacks on reduced SIMON versions with dynamic key-guessing techniques,” *SCIENCE CHINA Information Sciences*, vol. 61, pp. 1–3, 2018.

[29] A. Garcia-de-Prado, G. Ortiz, and J. Boubeta-Puig, “COLLECT: COLLaborativE ConText-aware service oriented architecture for intelligent decision-making in the Internet of Things,” *Expert Systems with Applications*, vol. 85, pp. 231–248, 2017.

[30] V. Grosso, G. Leurent, F. X. Standaert, and K. Varıcı, “LS-designs: bitslice encryption for efficient masked software implementations,” in *Fast Software Encryption. FSE 2014*, C. Cid and C. Rechberger, Eds., vol. 8540 of Lecture Notes in Computer Science, pp. 18–37, Springer, Berlin, Heidelberg, 2014.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 8493795, 10 pages
https://doi.org/10.1155/2021/8493795

Research Article

Addressing Overfitting Problem in Deep Learning-Based Solutions for Next Generation Data-Driven Networks

Mansheng Xiao,[1,2] Yuezhong Wu,[1] Guocai Zuo,[3] Shuangnan Fan,[2] Huijun Yu,[1] Zeeshan Azmat Shaikh,[4] and Zhiqiang Wen[1]

[1]*School of Computer Science, Hunan University of Technology, Zhuzhou 412007, China*
[2]*College of Electrical and Information Engineering, Hunan Institute of Traffic Engineering, Hengyang 421009, China*
[3]*School of Software and Information Engineering, Hunan Software Vocational and Technical University, Xiangtan 411100, China*
[4]*Department of Electrical Engineering, University of the Punjab, Lahore 54660, Pakistan*

Correspondence should be addressed to Yuezhong Wu; wuyuezhong@hut.edu.cn and Guocai Zuo; 474025986@qq.com

Received 27 May 2021; Revised 29 June 2021; Accepted 20 July 2021; Published 15 August 2021

Academic Editor: Dr. Muhammad Shafiq

Next-generation networks are data-driven by design but face uncertainty due to various changing user group patterns and the hybrid nature of infrastructures running these systems. Meanwhile, the amount of data gathered in the computer system is increasing. How to classify and process the massive data to reduce the amount of data transmission in the network is a very worthy problem. Recent research uses deep learning to propose solutions for these and related issues. However, deep learning faces problems like overfitting that may undermine the effectiveness of its applications in solving different network problems. This paper considers the overfitting problem of convolutional neural network (CNN) models in practical applications. An algorithm for maximum pooling dropout and weight attenuation is proposed to avoid overfitting. First, design the maximum value pooling dropout in the pooling layer of the model to sparse the neurons and then introduce the regularization based on weight attenuation to reduce the complexity of the model when the gradient of the loss function is calculated by backpropagation. Theoretical analysis and experiments show that the proposed method can effectively avoid overfitting and can reduce the error rate of data set classification by more than 10% on average than other methods. The proposed method can improve the quality of different deep learning-based solutions designed for data management and processing in next-generation networks.

1. Introduction

At present, the direction supported by the internet is changing from consumption to production, but the network architecture based on TCP/IP cannot adapt to this change in scalability, security, and other aspects. On the other hand, big data technology is also emerging in various industries. These emerging technologies are in the early stage of development, and there are still many problems to be solved [1–3]. In recent years, with the development of 5g technology, the amount of data stored in the computer system is increasing. Due to the diversity and uncertainty of massive data classification, there are a series of problems in data-driven network communication, such as the existence of these massive data not only occupies a lot of network space, but also causes network congestion [4, 5]. Therefore, how to classify and process the massive data before data transmission to reduce the amount of data transmission in the network is a very worthy problem [6–9]. The development of deep academic technology provides a good solution to this problem. At present, artificial intelligence technology based on deep learning has become more and more widely used in various fields. Convolutional neural network (CNN) is one of the core technologies of deep learning [4]. Compared with fully connected neural networks, CNN has features such as local connection, weight sharing, and downsampling can greatly reduce the complexity of the model and improve the training efficiency and data processing accuracy of the model. Therefore, it has been highly recognized by scholars in the field, and its application is gradually being promoted [10–14].

However, during the training process of the CNN model, due to the characteristics of the training data set itself, such as the data set is too small and contains a lot of noise, or the structure of the model itself, such as the model is too complex, the training parameters are too much, and the training is too much, the model is very likely to fall into overfitting [15–17]. For example, because the number of training samples is too small, the network parameters obtained after training cannot accurately simulate the distribution of all samples, or a large number of noise samples are fitted during the training process, ignoring part of the correct sample data, so the model fits the training set data very well. Good, but the poor fitting effect to the data outside the training set is reflected in the small loss function value (deviation) during training, while the loss function value is very large in verification or testing [18].

Since the development of machine learning technology, many experts and scholars have been conducting research on overfitting problems in the regression process and have achieved a series of research results. For example, Li et al. [19] proposed earlier to solve the overfitting problem in the algorithm, and Hinton et al.'s literatures [20, 21] first proposed the dropout method in machine learning to solve the overfitting problem. Chen et al. [22] adjusted algorithm parameters and iterations through singular value decomposition (SVD) to avoid the occurrence of overfitting phenomenon, and other methods such as literatures [23–25] have also carried out related research. In view of the overfitting problem existing in the current deep learning CNN model, many scholars and engineering technicians are currently working on the research of this problem, and a series of solutions have been proposed [10, 11, 13, 16, 18, 26, 27], including data expansion enhancement, regularization, discarding, and early stop method. However, affected by the research environment, conditions, and scope of application, the related theories for the development of deep learning technology are not mature enough; these research results can be described as different, each has its own advantages, and there are certain limitations. For example, in response to problems such as long training time and overfitting of the CNN model, Gong et al. [16] proposed to improve the CNN based on the immune system, but the accuracy of this improved network model on the test set is not high (only 81.6%). In terms of processing training data sets, Yang et al. [10] proposed an attribute reduction algorithm based on visual ranging, which solved the network overfitting problem by introducing Bayesian weight factor distribution instead of CNN fixed weights, but its application is very limited. In terms of data enhancement, literature [11] is aimed at the overfitting problem of the deep learning breast mass detection algorithm for synthetic images, using synthetic mammograms for training data enhancement, which is a common method. In the training process of the model, literature [26] conducted a systematic study on the standard reinforcement learning agent (reinforcement learning agent), conducted a general discussion on reinforcement learning overfitting, and generalized it from the perspective of inductive bias. Research on the behavior of deep learning did not propose a unique solution to the problem of deep learning overfitting. Literature [18] proposed a model prediction averaging method based on dropout double probability weighted pooling, which effectively reduces the error rate and inhibits overfitting, but the convergence speed of the algorithm becomes slow after the introduction of double probability. In the CNN model design, literature [12] introduced the particle swarm optimization (PSO) algorithm to reduce the back propagation of the error, avoid the lag error and the image overcombination, and improve the convergence speed, but the PSO is introduced to the CNN weight update lack of theoretical basis. Other methods have more or less "side effects" while improving the efficiency of CNN's data processing [6].

Based on the advantages of these new technologies, this paper proposes a data-driven network architecture, which is aimed at solving the problem of massive data filtering and classification in the development of the emerging future network, and the specific contents are as follows: based on the full analysis of the characteristics of CNN and the current research on CNN model overfitting by scholars at home and abroad, this paper proposes an algorithm for the maximum pooling dropout and weight attenuation overfitting problem, and through the theoretical derivation, as well as the image data collected in the network for classification experiment comparison, it proves that this method can effectively avoid overfitting in the training process of CNN model and improve the generalization ability of the model. The convolution neural network overfitting prediction method proposed in this paper can classify the massive data in the network, reduce the amount of data transmission, and improve the communication efficiency [28–31].

The main work of this paper is as follows: First, it analyzes the problems existing in mass data transmission based on data-driven network architecture and proposes mass data classification method of CNN in deep learning technology. The second is to design a CNN overfitting prediction model for maximum pooling dropout and weight attenuation, so as to reduce the overfitting in model training and provide the classification accuracy of the model. Thirdly, three experiments are designed to verify the effectiveness of the model.

The structure of this article is as follows. The second part introduces convolutional neural network and related technologies. The third part introduces maximum pooling dropout and the CNN model of weight attenuation. After proposing the overfitting prediction method, the fourth part is about analysis of experimental results. The last part includes conclusion.

2. Convolutional Neural Network and Related Technologies

A CNN structure generally consists of a convolutional layer, a pooling layer, and a fully connected layer. The convolutional layer and the pooling layer are alternately connected. After the convolutional layer is calculated, the pooling layer starts to execute, and then, convolution and pooling are performed. In this way, the convolution operation and the pooling operation alternately perform the extraction of sample features, and then, the fully connected layer is used to classify the extracted features. Because CNN has the characteristics of local connection, weight sharing, and downsampling, it is compared with the general neural network, and the extraction process has fewer connection parameters (weights), fewer feature dimensions, and stronger representation capabilities, so it has better generalization [8]. However, in CNN model training, due to various reasons, overfitting often occurs. Overfitting means that the gap between training error and test error is too large. That is to say, the complexity of the model is higher than that of the actual problem, and the model performs well in the training set, but poorly in the test set.

Let x_i^l be the i-th feature map of the l-th layer of the input sample, $i = 1, 2, \cdots f^l$, f^l is the total number of feature maps of the l-th layer, $w^{l,k}$ is the k-th convolution kernel of the l-th layer, and $b^{l,k}$ is the l-th layer corresponding to the k-th convolution kernel. $z^{l,k}$ is the output of the k-th convolution operation of the l-th layer, and $f(x^l)$ is the activation function of the l-th layer. The operation process of the convolutional layer is expressed as follows:

$$z^{l+1,k} = \sum_{i=1}^{f^l} \operatorname{conv}\left(w^{l,k}, x_i^l\right) + b_i^{l,k}, \tag{1}$$

$$x^{l+1,k} = f\left(z^{l+1,k}\right). \tag{2}$$

Among them, the formula (1) indicates that the convolution matrix obtained by convolution between x_i^l and the corresponding convolution kernel $w^{l,k}$ is summed and the corresponding offset is added to obtain the input $z^{l+1,k}$ of the next layer$(l+1)$. After the activation function is processed, you can get the k-th output matrix $x^{l+1,k}$ of the $l+1$-th layer.

The next layer of the convolutional layer is the pooling layer. The pooling layer interprets the features of the convolutional layer, thereby reducing the feature dimension and network complexity. The pooling operation is as follows:

$$s_m^{l+1} = \operatorname{pool}\left(s_{m,1}^l, s_{m,2}^l, \cdots, s_{m,j}^l, \cdots s_{m,n}^l\right), \tag{3}$$

where $s_{m,j}^l$ represents the value of the j-th pooling unit in the m-th pooling area in the feature map obtained by the convolution operation (i.e., matrix $x^{l+1,k}$), l represents the pooling layer where it is located, and n is the number of pooling units in the pooling area. If $2 * 2$ pooling is used, then $n = 4$, pool(.) means pooling operation, and average pooling or maximum pooling is often used. Maximum pooling is widely used because it can retain its representative characteristics. This article uses the largest value pooling method.

After the pooling is completed, perform the full connection operation, calculate the loss function value, and determine whether to back propagate. The loss function value calculation is one of the important links in CNN. Its value is directly related to the efficiency and quality of the CNN. The cross-entropy function is commonly used, expressed as follows:

$$L = -\frac{1}{N}\sum_{i=1}^{N}(y_i \ln (o_i)). \tag{4}$$

Among them, N is the number of training samples, y_i is the actual label value of sample i, and o_i is the network output value of sample i.

3. Maximum Pooling Dropout and Weight Attenuation CNN Model

The previous section introduced the basic structure of the convolutional neural network and the calculation methods of each network layer. On the basis of the above, this section designs a maximum pooling dropout and weight attenuation CNN model in order to avoid excessive in the model training process and the occurrence of the fitting situation.

3.1. Maximum Pooling Dropout. As explained in Section 2, maximum pooling is a commonly used method in the CNN pooling layer. In order to avoid overfitting during model training, this paper introduces the maximum pooling dropout in the CNN pooling layer.

Suppose the retention probability of each pooling area in the feature map to be pooled is p, and the size of p can be manually adjusted according to the actual situation. Generally, it is set to $p = 0.5$. Then, the inhibition probability of each unit in the pooling area is $q = 1 - p$. At the same time, it is assumed that the unit values $(s_{m,1}^l, s_{m,2}^l, \cdots s_{m,n}^l)$ in each pooling area m of the l layer are rearranged in ascending order, that is, the unit value after the arrangement is as follows: $0 < d_{m,1}^l < d_{m,2}^l < \cdots < d_{m,n}^l$ (note: the semilinear activation function ReLU is used here to make the activation value of all units nonnegative, so the smallest $d_{m,1}^l > 0$); then, $d_{m,j}^l$ is selected as the maximum value of the entire pooling area. The output after pooling is as follows: all unit values $(d_{m,j+1}^l, d_{m,j+1}^l, \cdots, d_{m,n}^l)$ greater than $d_{m,j}^l$ are suppressed, and only values $(0, d_{m,1}^l, d_{m,2}^l, \cdots, d_{m,j}^l)$ less than or equal to $d_{m,j}^l$ are suppressed retained, because of the maximum pooling (take the maximum value) among these retained values, the final output value of pooling is $d_{m,j}^l$, and the probability of its occurrence is $d_{m,j}^l$, that is,

$$p_j = \operatorname{poss}\left(s_m^{l+1} = d_{m,j}^1\right) = pq^{n-j}, j = 1, 2, \cdots n. \tag{5}$$

In the above formula, p_j is the probability of the reserved output of the m-th pooled cell in the j-th pooled region, which is the product of the reserved probability of the entire pooled region and the suppressed probability of the suppressed cell, and n is the number of cells in the pooled region. If the pooled region is $3 * 3$, then the number of pooled units $n = 9$. Analyzing formula (5), it can be seen that when the maximum value pooling dropout is performed in the pooling area, the j-th activation value $d_{m,j}^{l}$ in the pooling area is selected as the output value of the pooling area through polynomial arrangement, which is

$$s_m^{l+1} = d_{m,j}^{l}, p_j \in (p_1, p_2, p_3, \cdots, p_n). \tag{6}$$

Suppose there are r feature maps in layer l, each feature map has a size of s, and a pooling area size of n, if overlapping pooling is not considered, there are rs/n pooling areas, then the model parameter that the l layer may need to be trained is $f(j) = (j+1)^{rs/n}$ (plus a bias), that is, the number of model parameter that the maximum pooling may need to train is exponentially related to the number of pooling area units input to the pooling layer. Because the function $f(j)(j > 1)$ is an increasing function, so after introducing the maximum pooling dropout, the pooling unit is randomly suppressed, that is, j is reduced, and the parameter value of the model to be trained is reduced exponentially, which effectively reduces the complexity and thus can more effectively suppress overfitting.

3.2. Weight Attenuation Regularization Method. When training a large CNN, in addition to using the abovementioned maximum pooling dropout suppression unit to avoid overfitting and if the function value changes drastically in some areas of the model, it means that the parameter value (weight) of the function is too large, making the area. The absolute value of the derivative value is large, and the model becomes complicated. The weight attenuation restricts the norm of the parameter so that it cannot be too large, so as to reduce the complexity of the model and reduce the influence of noise input, thereby reducing the occurrence of excessive fitting, and this method is also called regularization method.

Suppose that the loss function L_0 of the model is shown in formula (4). When calculating the weight attenuation, a penalty term is added to the original loss function L_0, namely,

$$L = L_0 + \frac{\lambda}{2} \sum_{w} \|w\|^2. \tag{7}$$

In formula (7), L_0 is the original loss function (see formula (4)), w is the network weight, that is, the connection coefficient of neurons, and $\lambda(\lambda > 0)$ is the penalty coefficient, which is used to measure the ratio of the penalty to L_0 relationship, and 1/2 is designed for the convenience of derivation. The above penalty term is the sum of the squares of the network weight w. Taking the derivative of (7), we get the following:

$$\begin{cases} \dfrac{\partial L}{\partial w} = \dfrac{\partial L_0}{\partial w} + \lambda w, \\ \dfrac{\partial L}{\partial b} = \dfrac{\partial L_0}{\partial b}. \end{cases} \tag{8}$$

In formula (8), b is the bias of the network neural unit (such as the convolution kernel), which is included in the o_n of L_0, that is, $o_n = w \cdot x_{n-1} + b$. From the above formula, it can be found that after the penalty term is added, it has no effect on the update of the bias b, but for the weight value$w$,

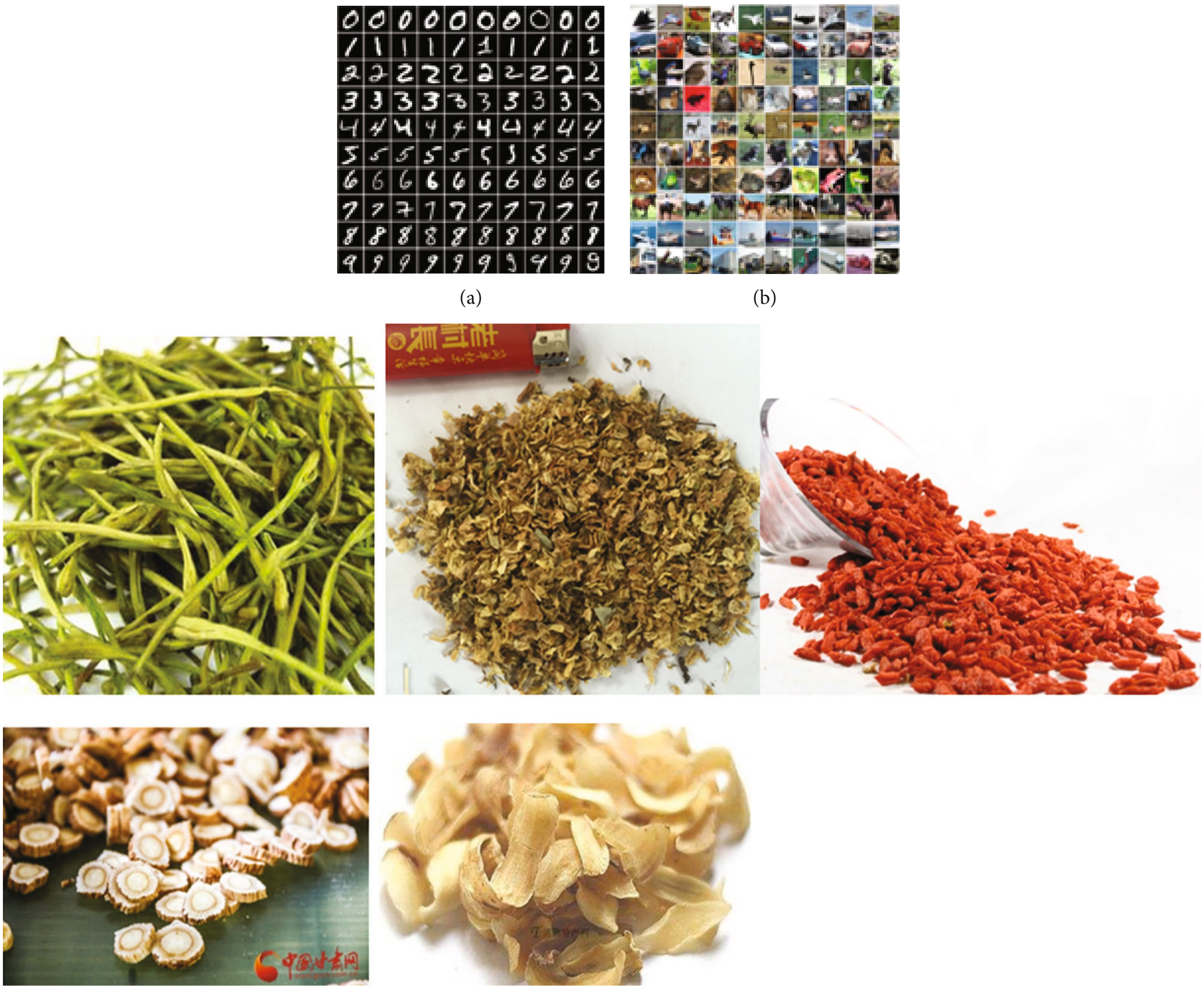
$$w' = w - \eta\left(\frac{\partial L_0}{\partial w} + \lambda w\right) = w - \eta\frac{\partial L_0}{\partial w} - \eta\lambda w = (1 - \eta\lambda)w - \eta\frac{\partial L_0}{\partial w}. \tag{9}$$

Among them, η is the learning rate. By analyzing formula (9), it can be seen that the coefficient of the weight value w is 1 before the penalty term is introduced, that is, $w' = w - \eta(\partial L_0/\partial w)$. After the penalty term is added, the coefficient before w becomes $1 - \eta\lambda$, because both η and λ are positive numbers less than 1, so 1-$\eta\lambda < 1$, that is, the effect of formula (9) is to reduce the value of w, which is the theoretical meaning of weight attenuation. Note that the $\eta(\partial L_0/\partial w)$ term in formula (9) is the gradient of the weight change of backpropagation. The expression is the same regardless of whether the penalty term is added or not. Therefore, the weight attenuation term mentioned here does not include this term. For further analysis of formula (9), when w is positive, the updated w' becomes smaller, and when w is negative, the updated w' becomes larger. Because of $|w| < 1$ (the weight after the network is normalized) in formula (9), the effect is to make w close to 0, that is, $|w| \longrightarrow 0$, to make the value of w as small as possible, which is equivalent to reducing the weight of the network, reducing the complexity of the network, and avoiding overfitting. It should be pointed out that the setting of the value of parameter λ in formula (9) is very important. If λ is too large, the weight w decreases too fast, underfitting may occur, or even training may not be possible, and if λ is too small, overfitting may occur. Together, the λ size setting can be adjusted based on the Bayes decision rule. This method assumes that the weights and biases of the network are random variables with specific distributions and are automatically calculated by statistical methods. For details, please refer to literature [32] (due to space limitations, there are no more details).

4. Experimental Results

4.1. Experimental Setup. Network training adopts Stochastic Gradient Descent, SGD, batch size = 100, initial learning rate $\eta = 0.1$, and when the error tends to be flat, reduce η and use a normal distribution with a mean of 0 and a variance of 0.01 to initialize the weight w; the bias is initialized to 0, and the probability $p = 0.5$ is retained by default. The method of parameter λ in the experimental penalty item of the method proposed in this paper is carried out according to the method described in Section 3.2, and experiment $\lambda = 0.2$.

Experimental environment: Win10, Intel Core i7 CPU@3.00, RAM 16GB, GPU NVIDIA GTX 1080Ti. *Experimental data set*: considering that CNN is very suitable for processing matrix data in digital images, and the image data in the network has a wide range of sources and a large



(a) (b)

(c)

Figure 1: Partial samples of (a) MNIST, (b) CIFAR-10, and (c) Chinese herbal medicine data sets.

amount of data, this paper selects 3 image data sets for experiments [30]. It is handwritten digit recognition data set MNIST, CIFAR-10, and Chinese herbal medicine identification data set. The first two data sets are currently commonly used benchmark data sets for testing the performance of CNN in the field of computer vision and deep learning. Because these two data sets exist in many websites, they have been tested by overfitting, identification, and classification to verify the effectiveness of the scheme, which is widely representative. The third data set is based on the massive Chinese herbal medicine data set freely collected by network communication technology. Some samples of these 3 data sets are shown in Figure 1. Among them, the MNIST data set contains 60,000 training samples and 10,000 test samples. Each sample is a 28 ∗ 28 single-channel grayscale image, containing 0-9 handwritten digits, which are normalized to [0, 1] before entering the CNN network. CIFAR-10 is a data set containing 60,000 natural images in 10 categories, including 50,000 training samples and 10,000 test samples. Each sample is a 32 ∗ 32 ∗ 3 RGB image, which is also normalized before input to the interval [0,1]. A total of 108000 images of Chinese herbal medicine data set were collected from the internet, including five categories of lily, Codonopsis pilosula, wolfberry, Sophora japonica, and honeysuckle, and these images were processed into the same size (224 ∗ 224) images and then normalized to the [0,1]. At the same time, in order to verify the performance and effect of the method proposed in this paper, the CNN method is based on the improvement of the immune system (referred to as ICNN) proposed in literature [16], the dropout method proposed by Hinton et al. in literature [21] (referred to as H_Dropout), and the random pooling method (abbreviated as SP) proposed in [33] and the method in this paper for comparison experiments, in which ICNN proposed an improved CNN network method based on the immune system for problems such as overfitting, although this method is tested in the data set. The accuracy is not high (only 81.6%), but its performance is stable, and the theoretical basis is sufficient; Hinton et al.'s H_Dropout method is a dropout method that combines a fully connected layer. They are the first scholars to use dropout to avoid CNN overfitting, and the method is mature. It is easy to implement, while the random pooling

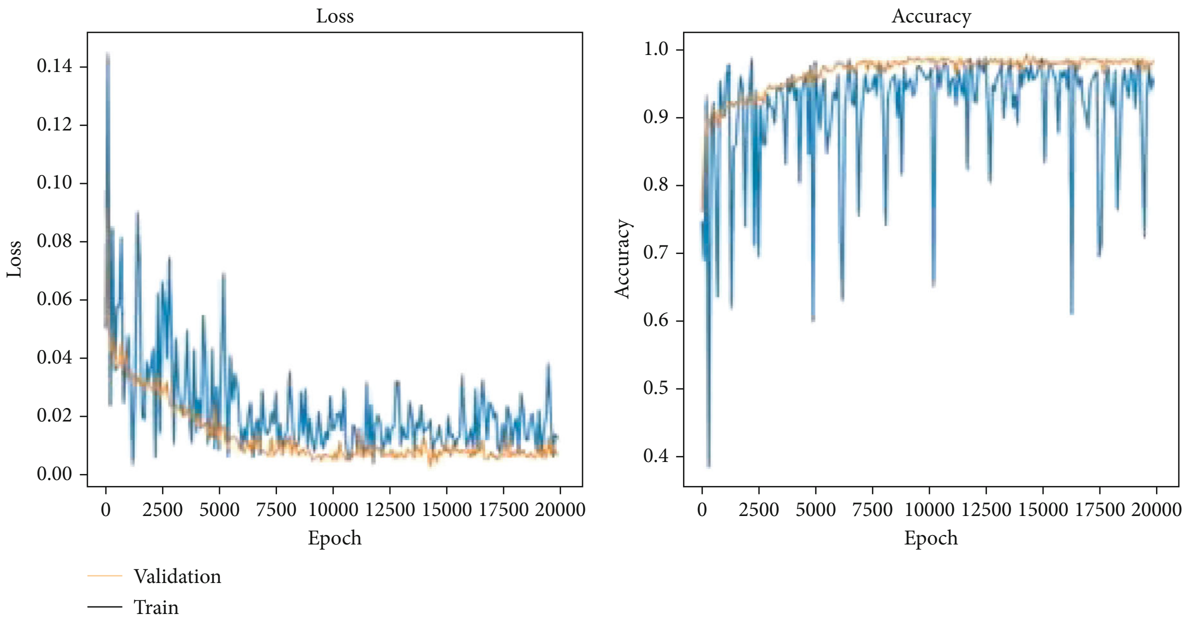


Figure 2: MNIST data set: the relationship between the number of iterations and the loss value and accuracy rate.

Table 1: Comparison of error rate (%) and retention probability of 4 methods in MNIST data set.

Retention probability	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
ICNN	6.36	6.36	6.36	6.36	6.36	6.36	6.36	6.36	6.36
H_Dropout	7.27	6.85	6.50	6.32	6.30	6.31	6.38	6.40	6.51
SP	2.23	2.17	2.10	2.08	2.13	2.26	2.31	2.37	2.49
MDWS	2.21	2.15	2.03	1.95	1.63	1.88	2.19	2.26	2.27

method proposed in [29] randomly selects the pooling activation value in training and uses the probability of each unit in the pooling area as the model average of the weighted probability during the test, which is a kind of efficiency. It is a high method to avoid overfitting. Therefore, the above three methods are feasible for comparison experiments. For convenience, the maximum pooling dropout and weight decay method proposed in this paper are referred to as MDWS (Maxpooling Dropout and Weight Scaled) for short. The experiment verifies the effectiveness of the method proposed in this paper by comparing the relationship between the loss function of the training set and the verification set, the iterative curve of the correct rate, and the error rate and retention probability in the test set.

4.2. Experiment 1: MNIST Data Set. In the experiment, the CNN model used in the method proposed in this paper is as follows: $1 * 28 * 28 \longrightarrow 6C5 \longrightarrow 2P2 \longrightarrow 12C5 \longrightarrow 2P2 \longrightarrow 1000N \longrightarrow 10N$: $1 * 28 * 28$ means the input is $28 * 28$ 1 channel image, C means Convolution, 6C5 means that the convolution kernel is $5 * 5$, and it contains a convolution layer with 6 convolution kernels (6 channels). *P* means pooling. The first “2” in the pooling layer 2P2 means the pooling step size. The “2” at the back means that the pooling core size is $2 * 2$, and 1000N means that the fully connected layer contains 1000 neurons. After calculation, the first fully connected layer should be 1152 neurons. The CNN network structure of the other counterparts (ICNN/H_Dropout/SP) is given in the relevant literature and will not be described in detail here. Figure 2 shows the relationship between the loss function value, the correct rate, and the iteration rounds of the method proposed in this article when the iteration round epoch = 20000. It can be seen from the figure that as the number of iterations increases, the loss of the training set, the value of the function keeps decreasing, and the accuracy rate keeps increasing. When the number of iterations exceeds 10,000 times, the value change tends to be stable. Similarly, the loss function value of the verification set also decreases with the increase of iteration rounds, and the accuracy rate continues to increase, and it stabilizes after 10,000 iterations, indicating that the method proposed in this paper can avoid the occurrence of overfitting.

Table 1 is a comparison of the average error rates of the four methods tested in their respective network models under different retention probabilities. Since the ICNN method has nothing to do with the retention probability, the ICNN method is only used to compare the error rate of the test without considering the impact of the retention probability. Analyzing the data in Table 1, the three methods (H_Dropout/SP/MDWS) have the lowest error rate when the retention probability is about 0.5, and the error rate of the H_Dropout method changes most drastically with the value of p. In addition, looking at the four methods under the same p value, the MDWS method proposed in this paper has the smallest error rate, such as 1.63% when p =0.5, SP method has a minimum error rate of 2.08% when p =0.4, and H_Dropout method when p =0.5. The error rate is 6.3%, which shows that the method in this paper can better reduce overfitting and have better pan-China.

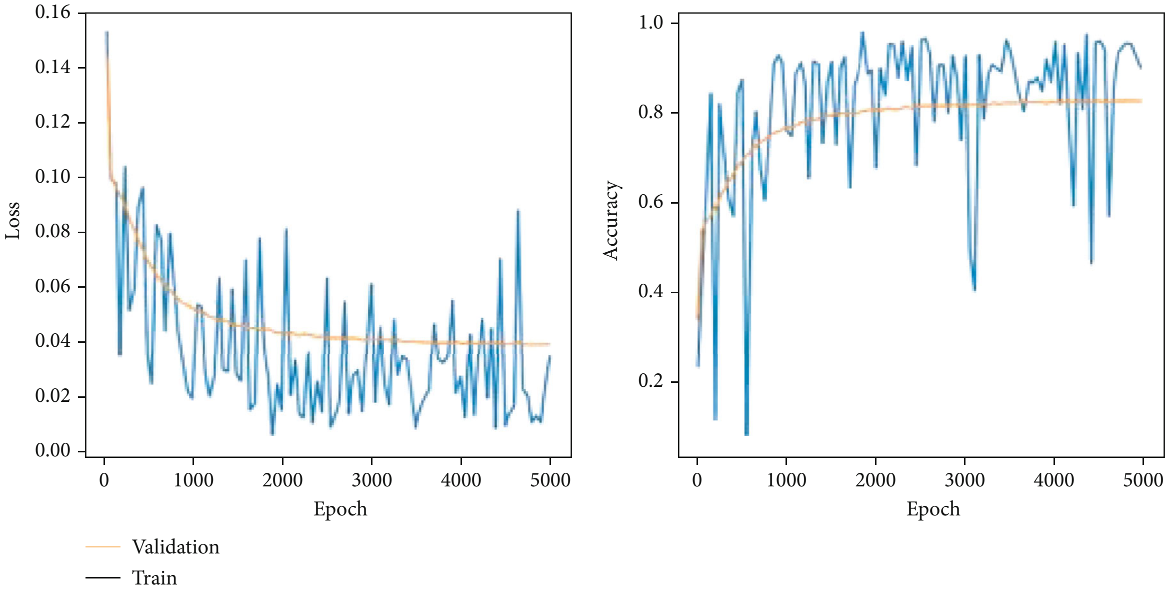


Figure 3: CIFAR-10 data set: the relationship between the number of iterations and the loss value and the accuracy rate.

Table 2: The relationship between the error rate and retention probability of the CIFAR-10 data set.

Retention probability	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
ICNN	7.44	7.44	7.44	7.44	7.44	7.44	7.44	7.44	7.44
H_Dropout	8.87	7.95	7.76	7.12	6.15	7.21	7.38	8.12	8.69
SP	4.83	4.67	4.55	4.31	4.09	4.02	4.16	4.47	5.39
MDWS	3.21	3.08	2.97	2.93	2.86	2.88	3.08	3.16	3.35

4.3. *Experiment 2: CIFAR-10 Data Set.*

This data set is a natural color image data set containing 3 channels. Compared with MNIST, the difference in each category is greater. For this reason, we designed the following CNN network structure: $3 \times 32 \times 32 \longrightarrow 32C5 \longrightarrow 3P2 \longrightarrow 128C3 \longrightarrow 3P2 \longrightarrow 2000N \longrightarrow 10N$; the interpretation of the model is the same as the MNIST data set. The other three methods are also carried out in accordance with the relevant requirements of the literature. The training is iterated for 5000 rounds in total, and the relationship between the loss function, the correct rate, and the iteration rounds of the training set and the validation set is shown in Figure 3. Analyzing Figure 3, it can be seen that as the number of iterations increases, the loss value of both the test set and the verification set continues to decrease, and the change gradually stabilizes. Similarly, as the accuracy value increases with the epoch, the training and verification curves are gradually rising and tending to be flat. It shows that the method proposed in this paper can effectively avoid overfitting when training on the CIFAR-10 data set.

Same as the experiment in Section 4.2, use the above 4 methods (ICNN, H_Dropout, SP, and MDWS in this article) to test in the respective trained networks with the test set and take different retention probability p values to obtain 4 methods under different retention probabilities. The error rate is shown in Table 2, where ICNN has nothing to do with the retention probability, which has been explained in Section 4.2. Analyzing the data in Table 2, it can be seen that the three methods have a lower classification error rate between the retention probability of 0.4 and 0.6. Among them, the H_Dropout method has the lowest error rate of 7.12% when $p = 0.4$, the SP method has the lowest error rate of 4.02% when $p = 0.6$, and the method in this paper has the lowest error rate of 2.86% when $p = 0.5$. In addition, from the entire table under the same retention probability of the four methods, the MDWS method proposed in this article has the lowest error rate (although the ICNN method does not retain the concept of probability, its classification error rate is higher than the method proposed in this article), In addition, from the entire table under the same retention probability of the four methods, the MDWS method proposed in this article has the lowest error rate (although the ICNN method does not retain the concept of probability, its classification error rate is higher than the method proposed in this article). This shows that the method proposed in this paper also has good generalization in the CIFAR-10 data set. The method is also available in the CIFAR-10 data set and has achieved good generalization.

4.4. *Experiment 3: Chinese Herbal Medicine Identification Data Set.*

The data set is a large-scale data set freely collected from the internet by using modern network technology, and the basic introduction has been stated before. There are many kinds of Chinese herbal medicine, even the same kind of Chinese herbal medicine will appear different forms due to the influence of growth environment and other factors, and some different kinds of Chinese herbal medicine are very similar. Compared with the previous two data sets, the image difference of each category in Chinese herbal medicine data set is small, so the classification processing is more difficult. Since each image in the

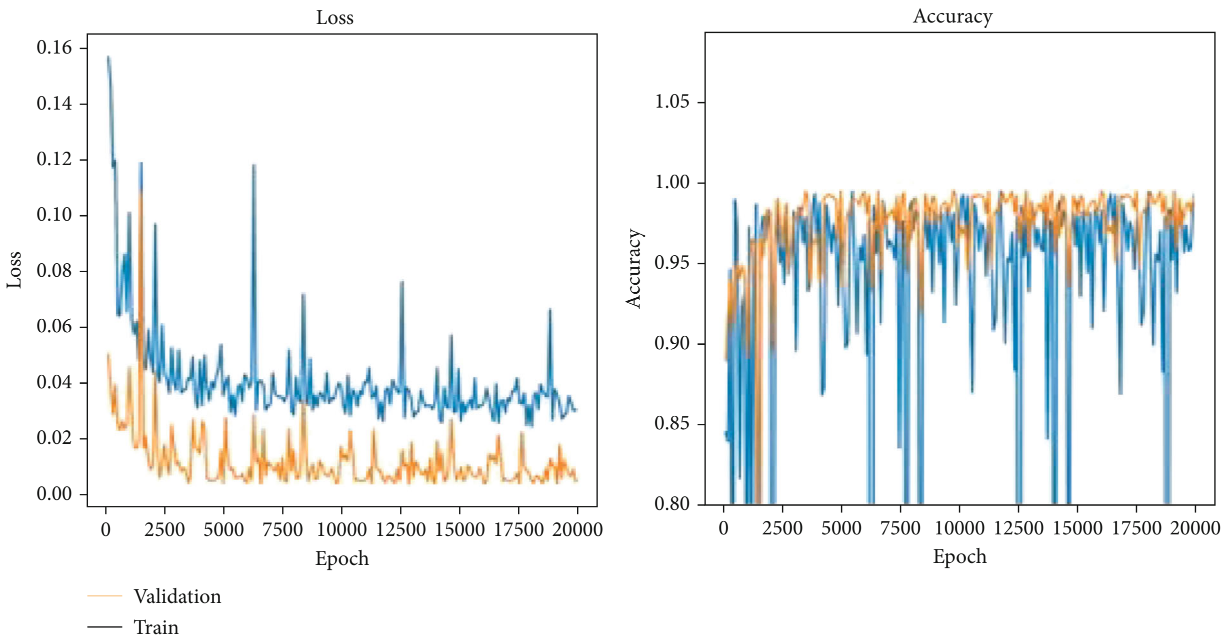


Figure 4: Chinese herbal medicine data set: the relationship between the number of iterations and the loss value and the accuracy rate.

Table 3: The relationship between the error rate and retention probability of the Chinese herbal medicine data set.

p	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
ICNN	7.44	7.44	7.44	7.44	7.44	7.44	7.44	7.44	7.44
H_Dropout	8.87	7.95	7.76	7.12	6.15	7.21	7.38	8.12	8.69
SP	4.83	4.67	4.55	4.31	4.09	4.04	4.16	4.47	5.39
MDWS	3.21	3.08	2.97	2.93	2.76	2.88	3.08	3.16	3.35

data set is also composed of 3-channel natural color images, all the images are processed into $224 * 224$ size as input, so we designed the following CNN network structure: $3 \times 224 \times 224$➔64C3➔1P2➔128C3➔1P2➔256C3➔1P2➔512C3➔4096N➔5N. The other three methods were also carried out according to the relevant requirements of the literature. The total number of training iterations is 20000, and the loss function of training set and validation set and the relationship between accuracy and iteration epochs are obtained, as shown in Figure 4. It can be seen from the analysis of Figure 4 that with the increase of iteration rounds, the loss value of both test set and validation set is decreasing, and the change gradually tends to be stable after 3000 epochs. Similarly, with the increase of epoch, the training and validation curve tends to be flat at 3000 epochs. The results show that the proposed method can effectively avoid overfitting when training on Chinese herbal medicine data set.

As in the previous experiment, the above four methods (ICNN, H_ Dropout, SP, and MDWS) are tested in the trained network with the test set, and different retention probabilities are taken to obtain the error rates of the 4 methods under different retention probabilities, as shown in Table 3. ICNN has nothing to do with the retention probability, which has been explained previously. By analyzing the data in Table 3, it can be seen that the three methods have a lower classification error rate between the retention probability of 0.5~0.7. Among them, H_Dropout method has the lowest error rate of 6.15% when $p = 0.5$, SP method has the lowest error rate of 4.04% when $p = 0.6$, and the method proposed in this paper has the lowest error rate of 2.76% when $p = 0.5$. Comparing the minimum error rate of the three methods, the MDWS method proposed in this paper has the minimum error rate, and it shows that the method proposed in this paper also has good generalization in Chinese herbal medicine data set.

5. Conclusion

With the development of computer network technology, in the network communication based on data-driven, the amount of data in cyberspace is increasing. The existence of massive duplicate data brings great problems to network communication and security. Therefore, how to correctly identify the same or similar data in the network is the basis to solve this problem. The development of deep learning technology provides a new solution to this problem. But the overfitting problem in the deep learning network model is a common problem encountered in model training. In the design of the CNN model, this article designs a maximum pooling dropout method in the pooling layer and introduces weights in the backpropagation. The weight attenuation mechanism is used to reduce the complexity of the deep learning model, so as to avoid the overfitting of the deep learning model in training and improve the robustness of network data communication. There are two main innovations in this article. One is the design of the maximum pooling dropout, which uses the unit value sorting of the pooled area to design the unit (neuron) discarding method. The second is to introduce a penalty term in the backpropagation to design the weight attenuation. The implementation process: theoretical analysis and experimental comparison verify that the method in this paper can effectively avoid overfitting and improve the generalization performance of the network. However, the method proposed in this paper still has the following problems. First, when the maximum pooling dropout is used, the semilinear activation function (ReLU) of the front convolutional layer may cause nonmaximum output unit value 0, so that the neuron corresponding to the unit will not be updated in the future (the gradient is 0), that is, dead nerve; the second is that the introduced penalty term parameter λ and the maximum pooled dropout probability p should theoretically have a certain connection, but in this article, the two are not connected for research, and the solution of these problems will be the next research goal.

Data Availability

The two data sets used in this experiment are MINIST and CIFAR-10, which are classic open source data sets for deep learning network testing, and can be downloaded from related websites. The two data sets used in this experiment are downloaded from Baidu paddle platform: https://www.paddlepaddle.org.cn/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported in part by the National Key R&D Program of China under grant number 2019YFE0122600, in part by the key scientific research projects of Hunan Provincial Education Department under grant number 20A162, and in part by the Natural Science Foundation of Hunan Province under grant numbers 2021JJ50050, 2020JJ7007 and 2020JJ6088.

References

[1] X. Chen, C. Li, D. Wang et al., “Android HIV: a study of repackaging malware for evading machine-learning detection,” *IEEE Transactions on Information Forensics and Security*, vol. 15, no. 1, pp. 987–1001, 2020.

[2] G. Lin, S. Wen, Q. Han, J. Zhang, and Y. Xiang, “Software vulnerability detection using deep neural networks: a survey,” *Proceedings of the IEEE*, vol. 108, no. 10, pp. 1825–1848, 2020.

[3] Y. Xu, C. Zhang, G. Wang, Z. Qin, and Q. Zeng, “A blockchain-enabled deduplicatable data auditing mechanism for network storage services,” *IEEE Transactions on Emerging Topics in Computing*, 2020.

[4] X. Zhou, X. Xu, W. Liang, Z. Zeng, and Z. Yan, “Deep learning enhanced multi-target detection for end-edge-cloud surveillance in smart IoT,” *IEEE Internet of Things Journal*, 2021.

[5] Y. Xu, J. Ren, Y. Zhang, C. Zhang, B. Shen, and Y. Zhang, “Blockchain empowered arbitrable data auditing scheme for network storage as a service,” *IEEE Transactions on Services Computing*, vol. 13, no. 2, pp. 289–300, 2020.

[6] L. Qi, C. Hu, X. Zhang et al., “Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4159–4167, 2021.

[7] Y. Xu, Q. Zeng, G. Wang, C. Zhang, J. Ren, and Y. Zhang, “An efficient privacy-enhanced attribute-based access control mechanism,” *Concurrency & Computation Practice & Experience*, vol. 32, no. 5, pp. 1–10, 2020.

[8] X. Zhou, W. Liang, S. Shimizu, J. Ma, and Q. Jin, “Siamese neural network based few-shot learning for anomaly detection in industrial cyber-physical systems,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 8, pp. 5790–5798, 2021.

[9] Y. Xu, X. Yan, Y. Wu, Y. Hu, W. Liang, and J. Zhang, “Hierarchical bidirectional RNN for safety-enhanced B5G heterogeneous networks,” *IEEE Transactions on Network Science and Engineering*, 2021.

[10] X. Yang, X. Li, Y. Guan, J. Song, and R. Wang, “Overfitting reduction of pose estimation for deep learning visual odometry,” *China Communications*, vol. 17, no. 6, pp. 196–210, 2020.

[11] K. H. Cha, N. Petrick, A. Pezeshk, C. G. Graff, and B. Sahiner, “Reducing overfitting of a deep learning breast mass detection algorithm in mammography using synthetic images,” in *Progress in Biomedical Optics and Imaging-Proceedings of SPIE*, pp. 188–194, San Diego, CA, USA, March 2019.

[12] G. González, S. Y. Ash, R. San José Estépar, and G. R. Washko, “Reply to Mummadiet al.: overfitting and use of mismatched cohorts in deep learning models: preventable design limitations,” *American Journal of Respiratory and Critical Care Medicine*, vol. 198, no. 4, pp. 545–555, 2018.

[13] A. Ashiquzzaman, A. K. Tushar, M. R. Islam et al., “Reduction of overfitting in diabetes prediction using deep learning neural network,” in *IT Convergence and Security 2017. Lecture Notes in Electrical Engineering, vol 449*, pp. 35–43, Springer, Singapore, 2018.

[14] S. Zhang, Y. Gong, and J. Wang, “The development of deep convolutional neural networks and their applications in the field of computer vision,” *Chinese Journal of Computers*, vol. 42, no. 3, pp. 453–482, 2019.

[15] Y. Xu, C. Zhang, Q. Zeng, G. Wang, J. Ren, and Y. Zhang, “Blockchain-enabled accountability mechanism against information leakage in vertical industry services,” *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1202–1213, 2021.

[16] T. Gong, T. Fan, J. Guo, and Z. Cai, “GPU-based parallel optimization of immune convolutional neural network and embedded system,” *Engineering Applications of Artificial Intelligence*, vol. 62, no. 25, pp. 384–395, 2017.

[17] D. Liu and J. Liu, “Neural network model for deep learning overfitting problem,” *Journal of Natural Science of Xiangtan University*, vol. 40, no. 2, pp. 96–99, 2018.

[18] J. Cheng, G. Zeng, D. Lu, and B. Huang, “Dropout-based improved convolutional neural network model averaging method,” *Journal of Computer Applications*, vol. 39, no. 6, pp. 1601–1606, 2019.

[19] J. Li, G. Qin, X. Wen, and F. Hu, “Over-fitting in neural network learning algorithms and its solving strategies,” *Journal*

of Vibration, Measurement & Diagnosis, vol. 22, no. 4, pp. 260–264+320, 2002.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[21] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detector,” *Computer Science*, vol. 3, no. 4, pp. 212–223, 2012.

[22] D. Chen, Z. Yan, and H. Liu, “Overfitting phenomenon of SVD series algorithms in scoring prediction,” *Journal of Shandong University (Engineering Science Edition)*, vol. 44, no. 3, pp. 15–21, 2014.

[23] W. Shen, Y. Li, Z. Yang, X. Wang, and X. Ye, “Attribute reduction to prevent overfitting,” *Application Research of Computers*, vol. 37, no. 9, pp. 2665–2668, 2020.

[24] G. Qin and Z. Li, “Over-fitting of BP NN research and its application problem,” *Engineering Journal of Wuhan University*, vol. 39, no. 6, pp. 55–58, 2006.

[25] Z. Yang, F. Gao, S. Fu, M. Tang, and D. Liu, “Overfitting effect of artificial neural network based nonlinear equalizer: from mathematical origin to transmission evolution,” *Science China Information Sciences*, vol. 63, no. 6, pp. 82–97, 2020.

[26] C. Zhang, O. Vinyals, R. Munos, and S. Bengio, “A study on overfitting in deep reinforcement learning,” *Statistics*, no. 2, pp. 1–25, 2018.

[27] J. Wang, Z. Wang, and H. Wang, “Improved CNN algorithm based on PSO algorithm and dropout,” *Journal of Changchun University of Technology*, vol. 40, no. 1, pp. 26–30, 2019.

[28] C. Zhang, Y. Xu, Y. Hu, J. Wu, J. Ren, and Y. Zhang, “A blockchain-based multi-cloud storage data auditing scheme to locate faults,” *IEEE Transactions on Cloud Computing*, 2021.

[29] X. Yan, Y. Xu, X. Xing, B. Cui, Z. Guo, and T. Guo, “Trustworthy network anomaly detection based on an adaptive learning rate and momentum in IIoT,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6182–6192, 2020.

[30] X. Zhou, Y. Hu, W. Liang, J. Ma, and Q. Jin, “Variational LSTM enhanced anomaly detection for industrial big data,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 5, pp. 3469–3477, 2021.

[31] X. Yan, Y. Xu, B. Cui, S. Zhang, T. Guo, and C. Li, “Learning URL embedding for malicious website detection,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 10, pp. 6673–6681, 2020.

[32] D. J. C. Mac Kay, “Bayesian interpolation,” *Maximum Entropy and Bayesian Methods*, vol. 50, no. 3, pp. 39–66, 1992.

[33] M. D. Zeiler and R. Fergus, “Stochastic pooling for regularization of deep convolutional neural networks,” 2013, http://arxiv.org/abs/1301.3557.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 2803335, 10 pages
https://doi.org/10.1155/2021/2803335

Review Article

Channel Access-Based Joint Optimization of AoI and SINR under Attack: Game Theory and Distributed Approach

Yaoqi Yang,[1] Xianglin Wei,[2] Renhui Xu,[1] Laixian Peng,[1] Shuai Cheng,[1] and Lin Ge[1]

[1] *College of Communications Engineering, Army Engineering University of PLA, Nanjing 210000, China*
[2] *The 63rd Research Institute, National University of Defense Technology, Nanjing 210007, China*

Correspondence should be addressed to Xianglin Wei; wei_xianglin@163.com and Laixian Peng; lxpeng@hotmail.com

Received 28 May 2021; Revised 22 June 2021; Accepted 2 July 2021; Published 20 July 2021

Academic Editor: Dr. Muhammad Shafiq

This paper focuses on the joint optimization of the Age of Information (AoI) and Signal to Interference plus Noise Ratio- (SINR-) oriented channel access problem under attack in the Wireless Sensor Networks (WSNs). Firstly, to overcome the uncertain, dynamic, and incomplete information constrains, an active probability model and a controlling channel model are proposed for the sensors and the receiving end, respectively. Secondly, to ensure the AoI and SINR of the data generated by the sensors when transmitted under attack, one utility function based on average AoI and SINR is defined. Then, considering the distributed feature of the channel access process, the joint optimization problem is formulated under the game theory structure. Then, a distributed learning algorithm is proposed to reach the Nash Equilibrium (NE) of the game. Finally, simulation results have verified the correctness and effectiveness of the proposed method.

1. Introduction

1.1. Background. AoI (Age of Information) refers to the time interval between data's generations to its receiving end, and it is significantly different from the traditional concept of per-packet delay. Therefore, due to the capacity of representing the freshness of the data, lots of efforts have been made on AoI, which is significantly different from the traditional concept of delay. Up to now, plenty of researchers have devoted themselves to AoI's research from different aspects. To be specific, the queueing model's effect on the average AoI has been investigated in [1, 2]; the scenario about multihop transmission is considered in [3, 4]; the packet with losing situation is revealed in the calculation of AoI in [5]. In addition, more and more works concentrate on the goal that minimizes the average AoI, and the typical given approach is to adopt the strategy with weighted-sum average AoI.

Owing to a series of advantages of AoI, it can bring the unexpected benefit when used in some traditional cases. The scenario contains internet of things (IoT) and cyber physics systems (CPS) with edge-enabled storage or caching [6], where the transmitted data is time critical and requires high timeliness at the receiving end. Driven by the timeliness, AoI is utilized to measure the performance of the wireless transmission. For example, in mission-ciritcal industrial Wireless Sensor Networks (WSNs), data freshness is very critical to ensure a high-quality product manufacturing. In addition, SINR (Signal to Interference plus Noise Ratio) is also an important indicator for evaluating the performance of the network [7]. Therefore, to comprehensively optimize the effectiveness and reliability of the network, we consider a scenario where the sensors are undergoing the wireless channel access attack [8–14], and we focus on how to select the available channel to transmit the data generated by sensors while keeping the freshness of the data and maximizing the SINR as much as possible.

The wireless channels could be allocated in centralized or distributed manner. For the first one, though we can solve the above problem by setting a central controller to allocate the channel resources based on the each sensor's average AoI payoff, two factors limit the proposal's practicality. On the one hand, the calculation capacity requirement for the central controller is high, especially when the number of the sensors is large. On the other hand, the efficiency of the controller may not meet the need of the sensors when the scale of the network is very large; for example, the attack has influenced the available channel set, but the channel access strategy has not been sent to some sensors in time. Compared with the centralized decision-making (DM) manner, the distributed proposal has the following advantages. Firstly, the implementation cost is low, because there is no need to set a central controller. Secondly, each sensor could make the channel access decision by itself, promoting the efficiency of the channel selection. Thirdly, the decision-making process can adapt to the change of the attack quickly in dynamic scenarios.

However, it is a nontrivial task to solve the problem in a distributed manner. The following aspects hinder us from directly using of the referred method. Firstly, taking the practical application of the sensors into consideration, a sensor would not access the channel when there is no data to transmit. Therefore, how to capture this dynamic attribution of the sensor will be a challenge. Secondly, the attack progress is unpredictable and the available channel set for each sensor is unknown; how to perform the DM matching with the changing environment is also a problem. Thirdly, there is no information exchange for each sensor in the distributed manner, so the sensor does not know the chosen channel and the current state (accessing the channel or not) of other sensors. Therefore, the information attribution constraints, which are uncertain, dynamic, and incomplete, let the goal to solve the channel access under dynamic attack problem more challenging.

To tackle this tricky problem, game theory is a suitable framework to coordinate the behavior between different sensors [15]. To be specific, we formulate a game which is based on average AoI and SINR indicator under attacks in the WSNs, and a distributed learning algorithm is proposed to obtain the solutions.

1.2. Related Work. It is convenient to use game theory to model and analyze the routing and resource allocation problems in a competitive environment and especially in the security issues of the wireless network. To be specific, when some users want to access limited channel resource, how to select the channel to transmit is the key point for users. It should be noted that all users want to maximize their profits. Therefore, the relationship between users needs to be accurately described to calculate their revenue separately. Besides, game theory can model the complex relationships in a dynamic and iterative viewpoint, which would give the better assignment scheme by deriving the NE solution.

As shown in Table 1, the efforts in the channel access with game theory are classified by their optimization goals, solution or method, and the attack consideration. It can be seen that mean throughput is the main optimization indicator in the literature [7, 16–18], and transmission with errors or collision situations are discussed in [19, 20]. Besides, the whole network's utilities are optimized in [21–24]. It is obvious that the joint optimization of AoI and SINR has not been paid enough attention under attack, and the proposed solution or method should be in the distributed manner. In order to make up the above research gap, the joint optimization issue under attack is researched in this paper.

1.3. Main Work and Contributions. In this paper, our main contributions are threefold:

(i) The AoI and SINR-oriented channel access model under attack is formulated as an optimization problem, in which the transmission and controlling channels and the unknown nature of attacks are included to formulate the average AoI and SINR. Then, a game-based framework is established to curve the uncertain, dynamic, and incomplete information constrains

(ii) A distributed learning algorithm is proposed to derive the NE of the game, in which the channel access algorithm based on stochastic learning automata is put forward

(iii) To evaluate the performance of the proposed algorithm, simulation experiments are conducted to verify the correctness and effectiveness of the proposed algorithm

The remainder of this paper is organized as follows. In Section 2, we describe the system model and formulate the problem. Our proposed algorithm is then introduced in Section 3. Simulations about the performance of the proposed algorithm are detailed in Section 4, and Section 5 concludes the paper.

2. System Model

2.1. Network Model. In the WSN, N sensor nodes are deployed, and they can be represented by the set $\mathcal{S} = \{S_1, S_2, \cdots, S_N\}$. In addition, the active probabilities of the sensor nodes are also considered here. To be specific, $S_n = 1$ means the n-th node is active; otherwise, $S_n = 0$ represents the inactive status of the n-th node.Denote the set $\mathcal{B} = \{n \in N : S_n = 1\}$ as an arbitrary nonempty active sensor node set, and Γ as the set for all the active sensor nodes. At this time, the active probability of the WSN is $\mu(\mathcal{S})$, and it can be expressed as

$$\begin{aligned} \mu(\mathcal{S}) &= \mu(S_1, S_2, \cdots, S_N) = \prod_{n=1}^{N} p_n, \\ p_n &= \begin{cases} \beta_n, & S_n = 1, \\ 1 - \beta_n, & S_n = 0. \end{cases} \end{aligned} \tag{1}$$

Table 1: A comparison of channel access efforts based on game theory model.

Reference	Optimization goal	Solution/method	Attack consideration
[7]	Mean throughput	A distributed learning algorithm	×
[19]	Collision slots	The Lagrangian extreme value approach	×
[16]	Mean throughput	A distributed and online algorithm	×
[17]	Mean throughput	An EGT algorithm	×
[21]	Quality of service	The stochastic game theory tool set	×
[20]	Transmission and blocking probabilities	A CA algorithm	×
[22]	Channel access and resource allocation	Convex optimization	×
[18]	Mean throughput	A closed-form solution for SG	×
[23]	Network utility	A distributed algorithm	×
[24]	Spectral efficiency	An expectation maximization algorithm	×
This work	Average AoI	Reinforcement learning-based distributed scheme	

Meanwhile, the active probability for $\mathscr{B}$, which represents an arbitrary nonempty active sensor node set, is

$$\sum_{\mathscr{B}\in\Gamma}\mu(\mathscr{B}) = 1-\mu(\mathscr{B}_0), \tag{2}$$

where $\mu(\mathscr{B}_0)$ is the probability when all sensor nodes are inactive, and it can be calculated as

$$\mu(\mathscr{B}_0) = \prod_{n=1}^{N}(1-\beta_n). \tag{3}$$

2.2. Channel Model. In the WSN, owing to the dynamic and complexity transmission environment, the active probability of the sensor nodes and stability of the wireless channel are always changing. Moreover, the attacker can also deteriorate the availability of the wireless channels. In order to ensure the essential information exchange among the sensor nodes, two kinds of wireless channels models are adopted here, i.e. TC (transmission channel) and CC (controlling channel) [25]. To be specific, the TC is used to transmit ordinary data, and CC is responsible for exchanging some control information, such as channel-selection status, and node active probabilities. For ease of narration, denote $\mathscr{A}_n = \{a_1, a_2, \cdots, a_M\}$ as the available TC set for the n-th sensor node, where M is the number of the available TCs. When all the active sensor nodes finished the sensing process and transmitted the sensed data through the wireless channels, the channel access strategy of the n-th sensor node is $a_n \in A_n$, which means the channel a_n is selected by the n-th node to transmit the data.

2.3. Attack Model. An attacker can damage the performance of the WSN, such as QoS (quality of service). To be specific, at one certain attacking time slot, some available channels may become unavailable; at this time, the channel's stability and the reliability decreased. Then, due to the unavailable channels, the AoI and SINR performances of the sensed data in the WSN are influenced. However, given the limited attacking capacity of the attacker, which is consistent with [25], there is at least one available TC in the data transmission process; i.e., it is impossible for the attacker to make all the TCs unavailable at one attacking time slot, and CC is always reliable and available.

2.4. AoI Model. Here, we firstly consider the average AoI model for single node when there is no attack. Then, in the proposed scenario, where the average AoI for multisensor nodes under attack needs to be derived, the average AoI expression with a closed form is derived.

2.4.1. AoI for Single Node without Channel Attack. In order to determine the average AoI of the sensed data, the model based on queue theory is detailed here. The serving rule is FCFS (first come first serve), and the queue model is $M/M/1$. According to [26], the average AoI of the sensed data can be determined by

$$AoI_T = \lim_{T\longrightarrow\infty} A_T = \lambda\left(E[XT] + \frac{E[X^2]}{2}\right), \tag{4}$$

where λ denotes the incoming rate of the sensed data, i.e., data generating rate; $E[\cdot]$ is the operation for calculating expectation value; and X and T represent the stochastic variables for the sensed data's arrival time and system time, respectively.

Under the $M/M/1-$FCFS queue model, where the sensed data's generating rate subjects to the Poisson distribution, the serving rate, i.e., the sensed data transmission rate in the wireless channels, obeys the negative exponential distribution with parameter μ. Based on [26] the serving rate ρ can be calculated as

$$\rho = \frac{\lambda}{\mu}. \tag{5}$$

At this time, the average AoI of $M/M/1-\text{FCFS}$ queue model is [26]

$$\text{AoI} = \frac{1}{\mu}\left(1 + \frac{1}{\rho} + \frac{\rho^2}{1-\rho}\right). \tag{6}$$

2.4.2. AoI for Multinodes with Channel Attack. Given the fact that multiple sensor nodes can access one wireless channel at the same time, the distribution of arrival time X and system time T will change. To be specific, when τ sensor nodes select the same wireless channel to transmit the sensed data at the same time, the data generating rate of the n-th sensor node should be calculated as

$$\lambda_i = \frac{\lambda i}{\sum_{j=1}^{\tau} \lambda j}, \tag{7}$$

where λi is the data generating rate by the i-th sensor node and the channel serving rate μ is unchanged. Therefore, take (7) into (6), the average AoI of sensed data generated by the i -th sensor node is

$$\begin{aligned}\text{AoI}_i &= \frac{1}{\mu}\left(1 + \frac{\mu}{\lambda_i} + \frac{(\lambda_i/\mu)^2}{1-(\lambda_i/\mu)}\right)\\ &= \frac{1}{\mu}\left(1 + \frac{\mu}{\lambda i/\left(\sum_{j=1}^{\tau}\lambda j\right)} + \frac{\left(\lambda i/\left(\sum_{j=1}^{\tau}\lambda j\right)/\mu\right)^2}{1-\left(\lambda i/\left(\sum_{j=1}^{\tau}\lambda j\right)/\mu\right)}\right).\end{aligned} \tag{8}$$

Furthermore, when the WSN is under attack, the average AoI will change at the same time. In order to accurately curve the dynamic channel access relationships among the sensor nodes, the channel selection status, sensor node category, and time slots need to be jointly considered. To be specific, let $C(e,t)$ be the sensor node set which accesses the e-th channel at the t-th time slot. Since the $C(e,t)$ is related with the accessed channel and time slot, it would change with the launching of the attack at one particular attacking time slot. At this time, the average AoI of data, which is generated by the i-th sensor node, can be calculated as

$$\begin{aligned}\text{AoI}_i &= \frac{1}{\mu}\left(1 + \frac{\mu}{\lambda_i} + \frac{(\lambda_i/\mu)^2}{1-(\lambda_i/\mu)}\right)\\ &= \frac{1}{\mu}\left(1 + \frac{\mu}{\lambda i/\left(\sum_{j\in C(e,t)}\lambda j\right)} + \frac{\left(\lambda i/\left(\sum_{j\in C(e,t)}\lambda j\right)/\mu\right)^2}{1-\left(\lambda i/\left(\sum_{j\in C(e,t)}\lambda j\right)/\mu\right)}\right).\end{aligned} \tag{9}$$

2.5. SINR Model. When the i-th active sensor node selects the channel $a_i \in \mathscr{A}_i$ to transmit the sensed data, the SINR of the i-th active sensor node, which is from the arbitrary active node set $\mathscr{B}$, under the channel access strategy (a_i, a_{-i}), can be calculated as

$$\text{SINR}_i(\mathscr{B}, a_i, a_{-i}) = \frac{p_i d_i^{-\alpha}}{\sum_{j\in\mathscr{B}\setminus\{i\}:\, a_j=a_i} p_j d_{ij}^{-\alpha} + \sigma}, \tag{10}$$

where p_i is the transmitting power of the i-th sensor node, d_i is the distance between the i-th sensor node and its corresponding receiver, α is the path loss efficient, d_{ij} is the distance between the i-th and j-th sensor nodes, and σ means the environment noise. Therefore, the molecular represents the transmitting power of the sensed data; the denominator is the sum of the interference of other sensor nodes choosing the channel a_i and the environment noise.

2.6. Problem Formulation. The problem that needs to be solved is how to make each sensor node's own channel access strategy to jointly minimize the AoI and maximize the SINR when the WSN is under attack, i.e.,

$$\begin{aligned}&\min\left\{a \times AoI_i - b \times \text{SINR}_i\right\}\\ &\text{s.t.}\begin{cases} j \in C(e,t)\\ 0 \le t \le T\\ a_i \in \mathscr{A}_i\\ 1 \le e \le M\\ 1 \le i \le N\\ \mathscr{B} \in \Gamma,\end{cases}\end{aligned} \tag{11}$$

where a and b are the weighting factors to make AoI and SINR optimize in the same dimension. Note that it is difficult to directly use the typical method to solve the formulated problem (11), e.g., convex optimization, because the relationships of the channel selection results are relevant to time. Therefore, in order to make the nontrivial problem solvable, the problem in (11) needs to be reformulated with the game theory perspective, which is shown in (12).

Moreover, the payoff $R_i(\mathscr{B}, a_i, a_{-i})$ needs to be defined based on the optimization goal in (11) at first, i.e.,

$$R_i(\mathscr{B}, a_i, a_{-i}) = a \cdot \frac{1/(L(\mathscr{B})-1)\sum_{j\in B\setminus\{i\}:\, a_j=a_i}\text{AoI}_j}{\text{AoI}_i} + b \cdot \text{SINR}_i, \tag{12}$$

where the number of channels in set $\mathscr{B}$ is represented by $L(\mathscr{B})$. The numerator of the first item in R_i represents the channel competition effect on the average AoI among the sensor nodes which select channel a_i; and the denominator

```
Input: $K = \{1, 2, \cdots k_{\max}\}$: the iteration times set; $\mathscr{A}(n)$: the available channel set for the active sensor node; $q_{nd}(i) = 1/|\mathscr{A}_n|$: initial mixed strategy of each sensor node ($\forall n \in N, \forall d \in \mathscr{A}_n$); $\mathscr{B}(i)$: the active sensor node set in the current slot; $b$: the learning step size.
Output: $q_{nd}(k)$: the final mixed strategy of the active sensor node ($k \geq 1, \forall n \in N, \forall d \in A_n$); the AoI utility $AoI_i$; the SINR utility $SINR_i$.
1:  For the iteration time $i = 1 : k_{\max}$ do
2:      If the sensor node is inactive
3:      Do nothing, i.e. :
$q_{nd}(i+1) = q_{nd}(i)$
4:      Else
5:      Perform the SLA algorithm, i.e.
6:      Derive the normalized payoff $r_n(i) = R_n(i)/R_n^{\max}$ by (12)
7:          If $d = a_n(i.)$
8:
$q_{nd}(i+1) = q_{nd}(i) + br_n(i)(1 - q_{nd}(i))$
9:          Else
10:
$q_{nd}(i+1) = q_{nd}(i) - br_n(i)q_{nd}(i)$
11:         End
12:         End
13:         Record $AoI_i$ based on (9)
14:         Record $SINR_i$ according to (10)
15:  End
```

Algorithm 1. Channel access strategy for the sensor node under attack

of first item is the average AoI of the data generated by the i-th sensor node, and the second item means the weighted SINR value. At this time, as for $R_i(\mathscr{B}, a_i, a_{-i})$, the larger its value is, the smaller of the AoI value is and the larger of the SINR value is, where the sensed data is fresher and more reliable.

Note that all the sensor nodes prefer to minimize the average AoI and maximize the SINR of the data to be transmitted; their relationships are contended and noncooperated. At this time, we aim at maximizing the expectation value of the defined payoff, so (11) is equally transformed into (13), where the AoI and SINR are jointly optimized for the varying active sensor node set:

$$P_1 : \max_{a_i} E_{\mathscr{B}}[R_i(\mathscr{B}, a_i, a_{-i})] = \max_{a_i} \sum_{\mathscr{B}\in\Gamma} \mu(\mathscr{B}) R_i(\mathscr{B}, a_i, a_{-i}). \tag{13}$$

3. Game-Based Joint Optimization of AoI and SINR

3.1. Basic Idea. To jointly optimize the AoI and SINR performance, the minimizing problem is formulated in (11), while it is not applicable to the typical convex optimization approach. Then, the problem is equivalently transformed in the perspective of game theory, which aims at reaching the NE of the games in (13). Finally, based on the stochastic learning automata, one distributed algorithm is proposed to reach the NE by determining each sensor node's channel access strategy under attack.

3.2. Stochastic Learning Automata. To derive the NE of the reformulated problem in (13), one distributed-learning algorithm is adopted at first, which is mainly based on the stochastic learning automata [27, 28]. Then, combining the established models in Section 2 with the stochastic learning automata, the contents of the stochastic learning automata algorithm include the following steps:

(Step 1) All the inactive sensor nodes keep the current state and do nothing;

(Step 2) In the current time slot, the whole active nodes determine their channel access strategy based on the current payoff;

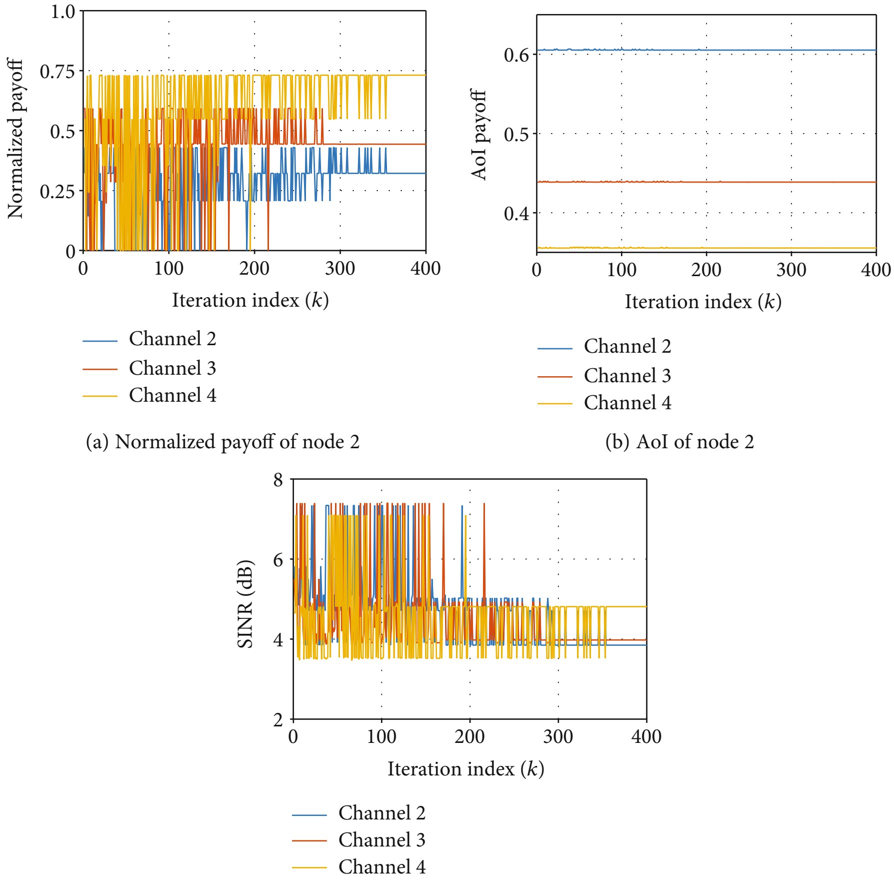
(Step 3) The channel access strategies are updated by the received payoff of the active sensor nodes at the next time slot.

3.3. Joint Optimization Algorithm. Note that TC can be used to transmit the sensed data, and the interactive information among the sensor nodes can be achieved by CC. Therefore, the sensor nodes can get their payoff instantaneously, which can be used to make the channel access strategy by itself in a distributed manner.

Based on the above analysis, the solution of the NE is detailed in Algorithm 1. To be specific, Steps 1–3 determine the channel access strategy for the active node; Steps 4–6 calculate the payoff of each sensor node; the channel access probability is included in Steps 7–8, where the payoff increased by choosing the current channel; Steps 9–12 decrease the channel access probability due to the decreased payoff; in Steps 13–15, the AoI and SINR utilities are determined finally.

Table 2: Parameter settings of the sensor nodes.

Node ID	Transmitting power	Generating rate	Available channel	Horizontal position	Vertical position
1	240	2	1, 2, 3, 4	53.78	20.39
2	630	3	2, 3, 4	49.11	226.60
3	255	4	1, 3, 4	544.70	328.04
4	175	5	3, 4	387.50	453.89
5	385	6	2, 3, 4	238.76	30.65
6	500	7	1, 2, 4	108.33	213.20
7	550	8	1, 2, 3, 4	318.55	411.87
8	300	9	1, 2, 4	371.37	361.69



(a) Normalized payoff of node 2

(b) AoI of node 2

(c) SINR of node 2

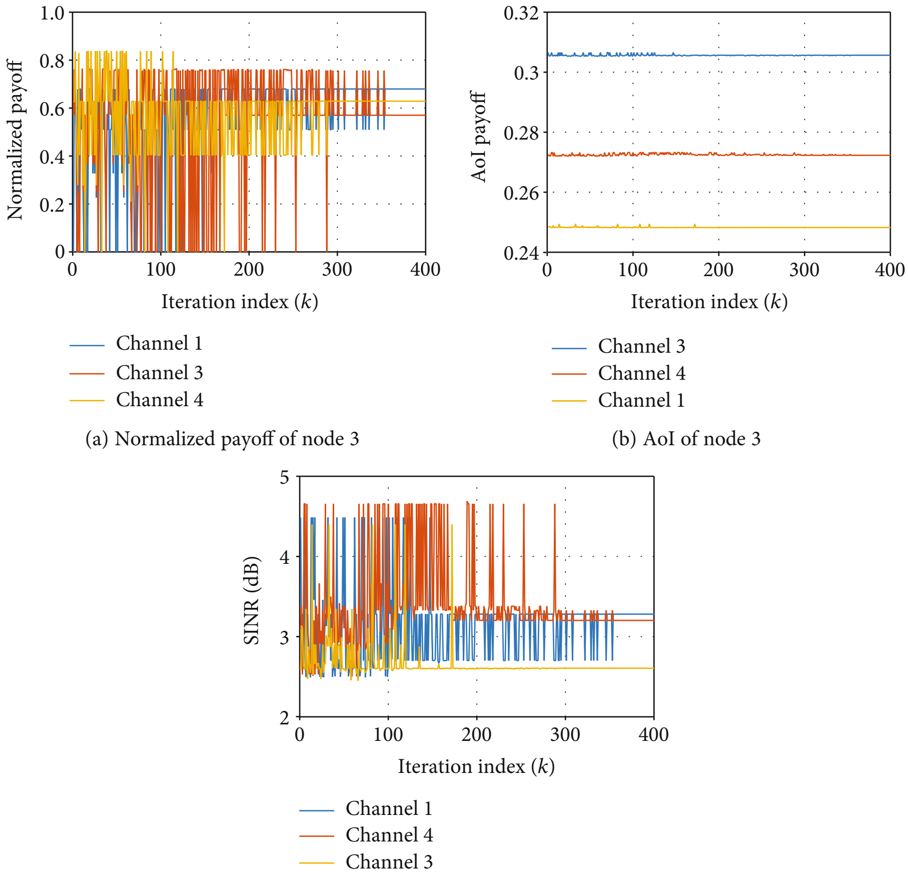
Figure 1: The performance of sensor node 2.

4. Simulation Results and Analysis

4.1. Parameter Settings. The simulation settings are listed in Table 2, where different sensor nodes have different transmitting powers, data generating rates, available channels, and positions. Besides, the serving rate of the wireless channel is set as 9.5, the active probability of the sensor nodes is 0.8, and the coefficients a and b are 5 and 1, respectively.

4.2. Compared Baselines. In order to evaluate the performance of the proposals, three algorithms are introduced as the baselines compared with the proposed algorithm.

(i) Optimal: the optimal algorithm is to find the best solution in a centric manner, which could can get the best performance by the exhausting searching approach.



(a) Normalized payoff of node 3

(b) AoI of node 3

(c) SINR of node 3

Figure 2: The performance of sensor node 3.

(ii) Best response: the best response algorithm could determine the best NE and worst NE of the game theory, which can be the upper and lower bounds of the solution in the game [29].

(iii) Random selection: the random selection algorithm means the sensor node select the channel to access, which has no relation with the defined payoff.

4.3. Correctness Verification. For ease of presentation, sensor nodes 2, 3, and 4 are selected as the example to show the correctness of the proposals.

4.3.1. Performance of the Sensor Node. Figure 1 is the normalized payoff, AoI, and SINR performance of the sensor node 2. As can be seen from Figure 1(a), with the iteration times increasing, the max payoff is obtained by selecting channel 4, the reason is that when node 2 selects channel 4, the AoI and SINR performance are the best, which are revealed in Figures 1(b) and 1(c). For ease of narration, sensor nodes 3 and 4 select channel 1 and channel 3 to transmit the sensed data, respectively, which are shown in Figures 2 and 3.

4.3.2. Channel Selection Probability. Figure 4 is the channel access probabilities of sensor node 2, sensor node 3, and sensor node 4. In Figure 4, on the one hand, when the iteration times increase, sensor node 2 would select channel 4, and so do channels 1 and 3 for nodes 3 and 4, respectively. On the other hand, in Figures 1, 2, and 3, the performance can reach best in the same channel selection results, so the correctness of the proposals is verified.

4.4. Effectiveness Verification. To evaluate the effectiveness of the proposed algorithm, 4 scenarios with heterogeneous active probabilities of the sensor nodes are considered here, where the active probabilities of the sensor nodes are set as follows:

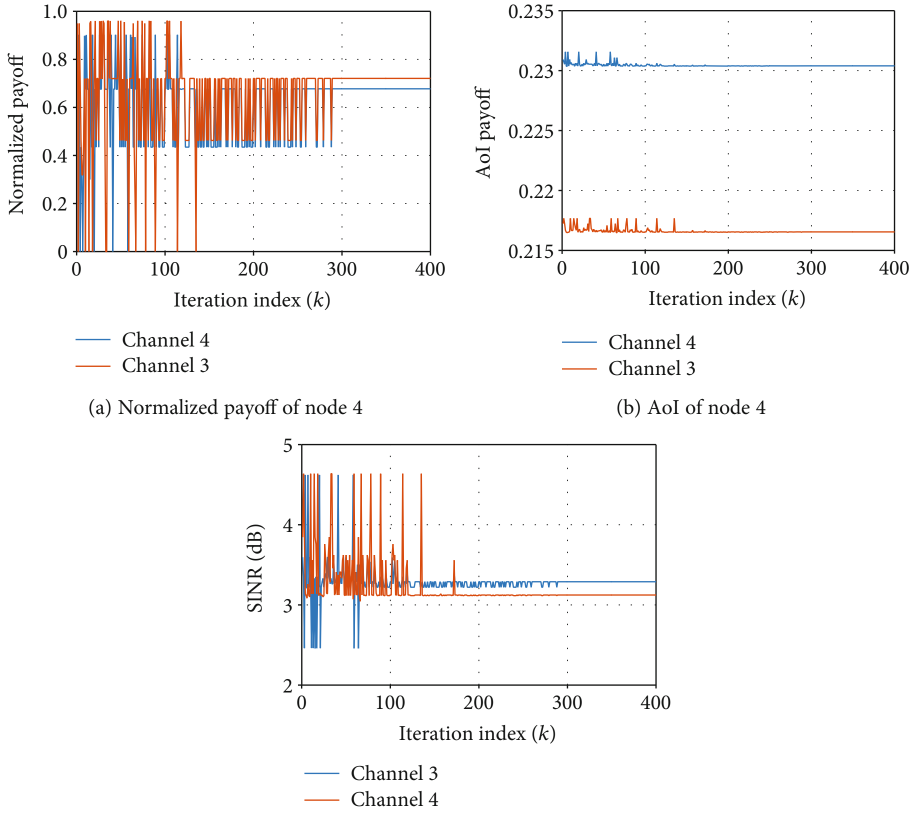
(Case 1) $\{0.1,0.2,0.3,0.5,0.7,0.9,0.8,0.9\}$

(Case 2) $\{0.2,0.3,0.4,0.6,0.8,0.9,0.9,0.9\}$

(Case 3) $\{0.3,0.5,0.6,0.8,0.9,0.9,0.9,0.9\}$

(Case 4) $\{0.6,0.6,0.8,0.9,0.9,0.9,0.9,0.9\}$

Figure 5 is the effectiveness performance verification under 4 scenarios with heterogeneous active probabilities. As shown in Figure 5, on the one hand, when the active



(a) Normalized payoff of node 4

(b) AoI of node 4

(c) SINR of node 4

Figure 3: The performance of sensor node 4.

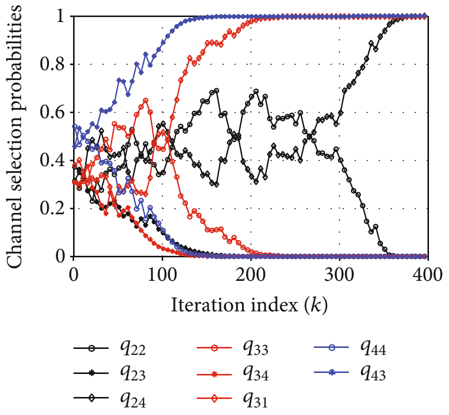


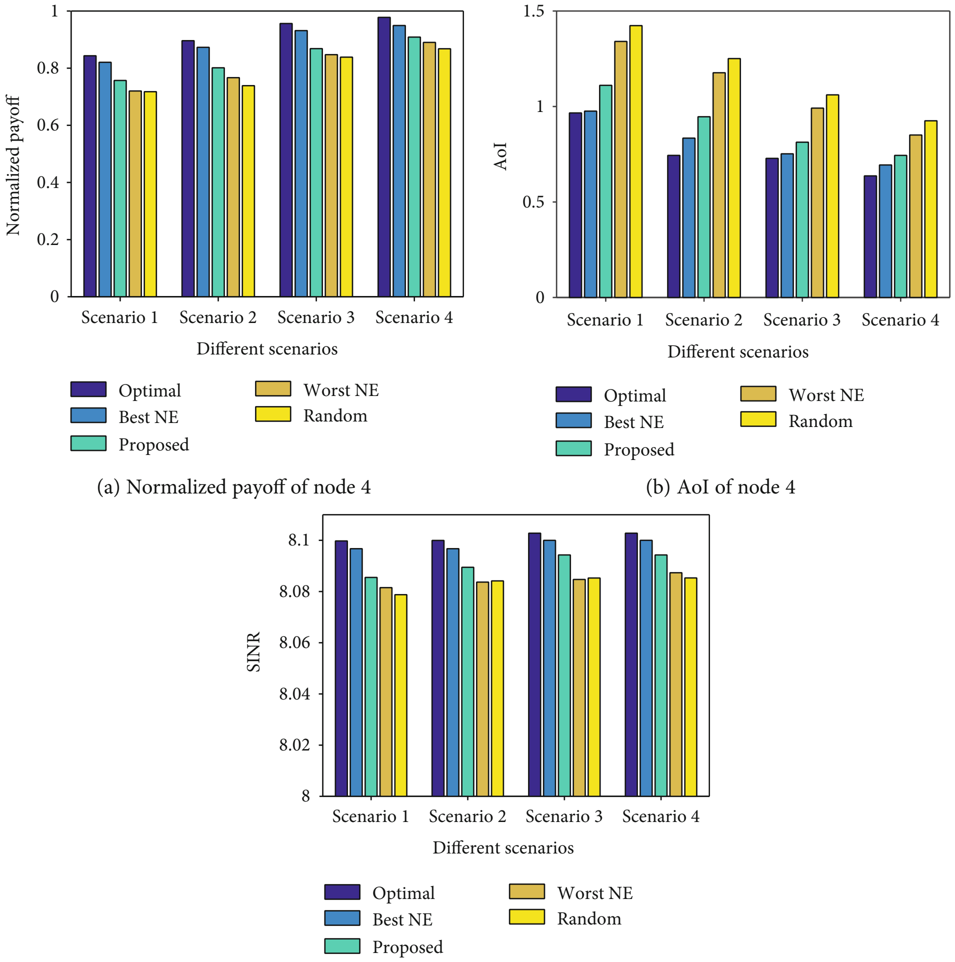
Figure 4: The channel access probabilities among different sensor nodes.

probability of the sensor nodes increases, the performance of the proposal is improved, this is because the probability of successfully accessing to the channel is positively related with the active probability of the sensor nodes; on the other hand, our proposal's performance is always better than the Worst NE and random selection scheme, which verifies the effectiveness of the proposals.

4.5. Discussion

4.5.1. Potential Application. In the Sixth Generation (6G) and Internet of Everything (IoE) era, with the rapid development of the wireless transmission technology, more and more data needs to be timely processed, especially in some time-critical networks. At this time, how to make plenty of wireless devices access within the limited wireless resources, e.g., channels, can be one desperate problem to be solved. Due to the consistent distributed attribution of the wireless devices, the channel access strategies proposed in this paper could be applied in the future, which can make the transmitted data keep fresh and reliable.

4.5.2. Attack Property. The attack property of the attacker is assumed to be stationary in this paper. In the next work, the channel access-based joint optimization of AoI and SINR under dynamic attack will be our focus. To be specific, inspired by the concept of time slicing network, the dynamic attack could be finished by launching attack at several different time slots. Combined with the joint optimization proposal at the particular time slot, the distributed channel access scheme under dynamic attack can be obtained by the method, where the dynamic attack process is divided into several attacking time slots.



(a) Normalized payoff of node 4

(b) AoI of node 4

(c) SINR of node 4

Figure 5: The effectiveness performance verification.

5. Conclusion

This paper proposed an algorithm to jointly optimize the AoI and SINR with channel accessing, when the WSN is under attack. Firstly, system models are established to derive the AoI and SINR indicator under attack. Then, the joint optimization problem is formulated from the perspective of game theory. To reach the NE of the game, one distributed algorithm is proposed next. Finally, simulation experiments are conducted to evaluate the correctness and effectiveness of the proposals. In the future, we will consider the joint optimization-based channel access issue under dynamic attacks.

Data Availability

No additional data is available in this paper.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 61671471).

References

[1] H. B. Beytur, S. Baghaee, and E. Uysal, “Towards AoI-aware Smart IoT Systems,” in *2020 International Conference on Computing, Networking and Communications (ICNC)*, pp. 353–357, Big Island, HI, USA, 2020.

[2] A. M. Bedewy, Y. Sun, and N. B. Shroff, “Age-optimal information updates in multihop networks,” in *2017 IEEE International Symposium on Information Theory (ISIT)*, pp. 576–580, Aachen, Germany, jan 2017.

[3] C. Kam, S. Kompella, G. D. Nguyen, and A. Ephremides, “Effect of message transmission path diversity on status age,” *IEEE Transactions on Information Theory*, vol. 62, no. 3, pp. 1360–1374, 2016.

[4] R. Talak, S. Karaman, and E. Modiano, “Minimizing age-of-information in multi-hop wireless networks,” in *2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pp. 486–493, Monticello, IL, oct 2017.

[5] C. Kam, S. Kompella, G. D. Nguyen, J. E. Wieselthier, and A. Ephremides, "Age of information with a packet deadline," in *2016 IEEE International Symposium on Information Theory (ISIT)*, vol. 2016, pp. 2564–2568, Barcelona, Spain, jul 2016.

[6] X. Wei, J. Liu, Y. Wang, C. Tang, and Y. Hu, "Wireless edge caching based on content similarity in dynamic environments," *Journal of Systems Architecture*, vol. 115, article 102000, 2021.

[7] Y. Xu, Y. Xu, and A. Anpalagan, "Database-assisted spectrum access in dynamic networks: a distributed learning solution," *IEEE Access*, vol. 3, pp. 1071–1078, 2015.

[8] W. Wang, H. Xu, M. Alazab, T. R. Gadekallu, Z. Han, and C. Su, "Blockchain-based reliable and efficient certificateless signature for IIoT devices," *IEEE Transactions on Industrial Informatics*, p. 1, 2021.

[9] J. Song, Q. Zhong, W. Wang, C. Su, Z. Tan, and Y. Liu, "FPDP: flexible privacy-preserving data publishing scheme for smart agriculture," *IEEE Sensors Journal*, p. 1, 2020.

[10] L. Zhang, Y. Zou, W. Wang, Z. Jin, Y. Su, and H. Chen, "Resource allocation and trust computing for blockchain-enabled edge computing system," *Computers and Security*, vol. 105, article 102249, 2021.

[11] W. Wang, H. Huang, L. Zhang, and C. Su, "Secure and efficient mutual authentication protocol for smart grid under blockchain," *Peer-to-Peer Networking and Applications*, no. article 1020, pp. 1–13, 2020.

[12] Z. Lejun, Z. Zhijie, W. Weizheng et al., "A covert communication method using special bitcoin addresses generated by vanitygen," *Computers, Materials and Continua*, vol. 65, no. 1, pp. 597–616, 2020.

[13] W. Wang and C. Su, "Ccbrsn: a system with high embedding capacity for covert communication in bitcoin," in *IFIP International Conference on ICT Systems Security and Privacy Protection*, pp. 324–337, Maribor, Slovenia, 2020, September.

[14] L. Zhang, M. Peng, W. Wang, Z. Jin, Y. Su, and H. Chen, "Secure and efficient data storage and sharing scheme forblockchain-based mobile-edgecomputing," *Transactions on Emerging Telecommunications Technologies*, no. article e4315, 2021.

[15] R. B. Myerson, *Game Theory: Analysis of Confict*, Harvard University Press, Cambridge, MA, USA, 1991.

[16] J. Zheng, Y. Cai, N. Lu, Y. Xu, and X. Shen, "Stochastic game-theoretic spectrum access in distributed and dynamic environment," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 10, pp. 4807–4820, 2015.

[17] M. A. Shattal, A. Wisniewska, A. Al-Fuqaha, B. Khan, and K. Dombrowski, "Evolutionary game theory perspective on dynamic spectrum access etiquette," *IEEE Access*, vol. 6, pp. 13142–13157, 2018.

[18] A. K. Lamba, R. Kumar, and S. Sharma, "Joint user pairing, subchannel assignment and power allocation in cooperative non-orthogonal multiple access networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 11790–11799, 2020.

[19] S. Gopal, S. K. Kaul, R. Chaturvedi, and S. Roy, "A non-cooperative multiple access game for timely updates," in *IEEE INFOCOM 2020 - IEEE conference on computer communications workshops (INFOCOM WKSHPS)*, pp. 924–929, Toronto, ON, Canada, 2020.

[20] A. Jella and S. L. Sabat, "Dynamic channel access of secondary users in a heterogeneous network using game theory," in *2018 10th International Conference on Communication Systems & Networks (COMSNETS)*, pp. 425–428, Bengaluru, 2018.

[21] L. Toka, M. Szalay, D. Haja, G. Szab, S. Rcz, and M. Telek, "To boost or not to boost: a stochastic game in wireless access networks," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pp. 1–6, Dublin, Ireland, 2020.

[22] A. Khodmi, S. B. Rejeb, N. Agoulmine, and Z. Choukair, "Joint user-channel assignment and power allocation for non-orthogonal multiple access in a 5G heterogeneous ultra-dense networks," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 1879–1884, Limassol, Cyprus, 2020.

[23] W. Yuan, P. Wang, W. Liu, and W. Cheng, "Variable-width channel allocation for access points: a game-theoretic perspective," *IEEE Transactions on Mobile Computing*, vol. 12, no. 7, pp. 1428–1442, 2013.

[24] Z. Wang, F. Yang, S. Yan, S. Memon, Z. Zhao, and C. Hu, "Joint design of coalition formation and semi-blind channel estimation in fog radio access networks," *China Communications*, vol. 16, no. 11, pp. 1–15, 2019.

[25] B. Wang, Yongle Wu, K. J. R. Liu, and T. C. Clancy, "An anti-jamming stochastic game for cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 877–889, 2011.

[26] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: how often should one update?," in *2012 Proceedings IEEE INFOCOM*, pp. 2731–2735, Orlando, FL, USA, mar 2012.

[27] K. Verbeeck and A. Nowe, "Colonies of learning automata," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 32, no. 6, article 772780, pp. 772–780, 2002.

[28] P. S. Sastry, V. V. Phansalkar, and M. Thathachar, "Decentralized learning of Nash equilibria in multi-person stochastic games with incomplete information," *IEEE Transactions on systems, man, and cybernetics*, vol. 24, no. 5, article 769777, pp. 769–777, 1994.

[29] D. Monderer and L. S. Shapley, "Potential games," *Games and economic behavior*, vol. 14, no. 1, article 1243143, pp. 124–143, 1996.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 9915408, 17 pages
https://doi.org/10.1155/2021/9915408

Research Article

Multilevel Privacy Controlling Scheme to Protect Behavior Pattern in Smart IoT Environment

Asad Khan,[1] **Muhammad Mehran Arshad Khan**,[2,3] **Muhammad Awais Javeed**,[4] **Muhammad Umar Farooq**,[5] **Adeel Akram**,[6] **and Chengliang Wang**[2]

[1]*School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou 510006, China*
[2]*School of Computer Science and Technology, Chongqing University, Chongqing 400044, China*
[3]*The Department of Examinations, GC University Faisalabad, Pakistan*
[4]*School of Information Engineering, Chang'an University, Xi'an 710064, China*
[5]*School of Computer Science and Technology, University of Science and Technology of China, China*
[6]*School of Information Engineering, Xuzhou University of Technology, China*

Correspondence should be addressed to Asad Khan; asad@gzhu.edu.cn
and Muhammad Mehran Arshad Khan; to_rabimehranrana@yahoo.com

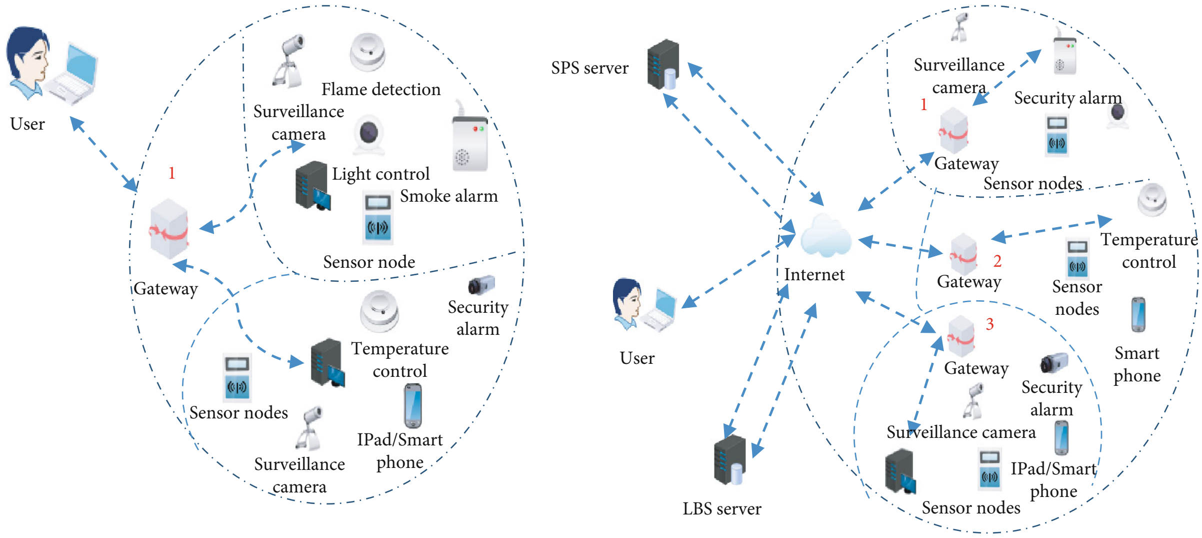
Received 1 April 2021; Revised 7 May 2021; Accepted 21 June 2021; Published 7 July 2021

Academic Editor: Muhammad Shafiq

Traditional approaches generally focus on the privacy of user's identity in a smart IoT environment. Privacy of user's behavior pattern is an important research issue to address smart technology towards improving user's life. User's behavior pattern consists of daily living activities in smart IoT environment. Sensor nodes directly interact with activities of user and forward sensing data to service provider server (SPS). While availing the services provided by a server, users may lose privacy since the untrusted devices have information about user's behavior pattern and it may share data with adversary. In order to resolve this problem, we propose a multilevel privacy controlling scheme (MPCS) which is different from traditional approaches. MPCS is divided into two parts: (i) behavior pattern privacy degree (*BehaviorPrivacyDeg*), which works as follows: firstly, frequent pattern mining-based time-duration algorithm (FPMTA) finds the normal pattern of activity by adopting unsupervised learning. Secondly, patterns compact algorithm (PCA) is proposed to store and compact the mined pattern in each sensor device. Then, abnormal activity detection time-duration algorithm (AADTA) is used by current triggered sensors, in order to compare the current activity with normal activity by computing similarity among them; (ii) multilevel privacy design model: we have divided privacy of users into four levels in smart IoT environment, and by using these levels, the server can configure privacy level for users according to their concern. Multilevel privacy design model consists of privacy-level configuration protocol (PLCP) and activity design model. PLCP provides fine privacy controls to users while enabling users to set privacy level. In PLCP, we introduce level concern privacy algorithm (LCPA) and location privacy algorithm (LPA), so that adversary could not damage the data of user's behavior pattern. Experiments are performed to evaluate the accuracy and feasibility of MPCS in both simulation and real-case studies. Results show that our proposed scheme can significantly protect the user's behavior pattern by detecting abnormality in real time.

1. Introduction

With the rapid advancement of sensor technology and mobile social networks, privacy of user's behavior pattern is becoming an essential part of smart IoT environment. Smart IoT environment typically consists of low power, resource restraint devices, and sensor nodes which are installed over the target region [1]. Sensor technology is associated to user's behavior pattern and human cognitive capture, which have been promoted in almost every smart IoT environment. Smart IoT environment typically consists of variety of embedded sensor nodes, actuators nodes, smart home local gateway, service provider sever (SPS), and users as shown in Figure 1. Personal smart home, business (sales track),



(a) Single gateway smart environment layout

(b) Multigateway smart environment layout

Figure 1: Layout of single and multigateway smart environment.

healthcare (cognitive behavior), and safety (military security and traffic management) are few fields with diverse applications. Furthermore, location-aware services, environmental monitoring, and architectural control are other appliances of smart IoT environment technology. During daily living activities, users interact with smart phones or tablets and can easily download many kinds of location-based server (LBS) applications and data from Google play store or Apple store by submitting their real location and related information to various LBS servers [2, 3]. Basically, if users want to avail the services of smart IoT environment, then they have to share some of their personal data to the service provider server (SPS) through local gateway sensor nodes and actuator devices. Although this kind of services makes daily life of users more comfortable, however, users enjoy these facilities in the smart IoT environment at the cost of their behavior pattern privacy [3]. For instance, users can easily search the location of any room or office by sending message with their location and query data to server through resource-restrained local home gateway [4]. Therefore, the server and low capacity smart IoT environment nodes (SHNs) can continuously access sensitive and personal data from users' requests and observe their personal information, such as their daily behavior pattern including what they do at certain time of a day [5]. More seriously, it can send private information to adversary which could then exploit privacy [6], such as user identity, user office's timing, occupation, home address, and user daily behavior activities. In smart IoT environment, once sensitive data are transmitted over the network, then it will be out of the user's control. All these appalling possibilities conflict with the privacy concerns of users' daily behavior pattern; therefore, we have to focus on users' behavior pattern privacy in a smart IoT environment.

There are two kinds of approaches, for collection of data, to detect abnormal activity: (i) video based and (ii) sensor based. Video-based approaches generally use technology of image processing; however, there are limitations in these approaches:

(i) Identifying the type of user's activity with small scope and small short time duration

(ii) Covering very small area and high cost

(iii) Violating user's privacy

Sensor-based approach is an emerging research area which has been adopted in smart IoT environment in order to tackle abovementioned pitfalls [7]. To some extent, it has been successfully used in smart IoT environment; however, they only process simple trajectory data and occasionally implement centralized data processing [8, 9]. Therefore, many of them have the following disadvantages.

(i) *Lack of Behavior Pattern Privacy*. They only focus on sequence information of activity and ignore important problem of preserving protection of user's behavior pattern privacy.

(ii) *Ignoring Time Duration in Location Privacy*. They ignore the use of time duration in order to detect duration abnormality. Furthermore, it did not consider combining location privacy and user's activity privacy in single approach.

(iii) *Computational Cost*. It consumes large bandwidth and uses centralized approach with long response time.

We focus to cover the abovementioned pitfalls and on protecting the user's behavior pattern privacy in smart IoT environment. The current study does not cover privacy edification of the whole system, and this research is an extension of privacy model. Our work is aimed at solving two main challenges in smart IoT environment, (i) ensuring privacy of user's behavior pattern, e.g., if a user is in a particular building from 9 : 00 to 14 : 00 and adversary can access this information, however, adversary cannot know where he/she was at 10 : 00 a.m. within the building and in which room

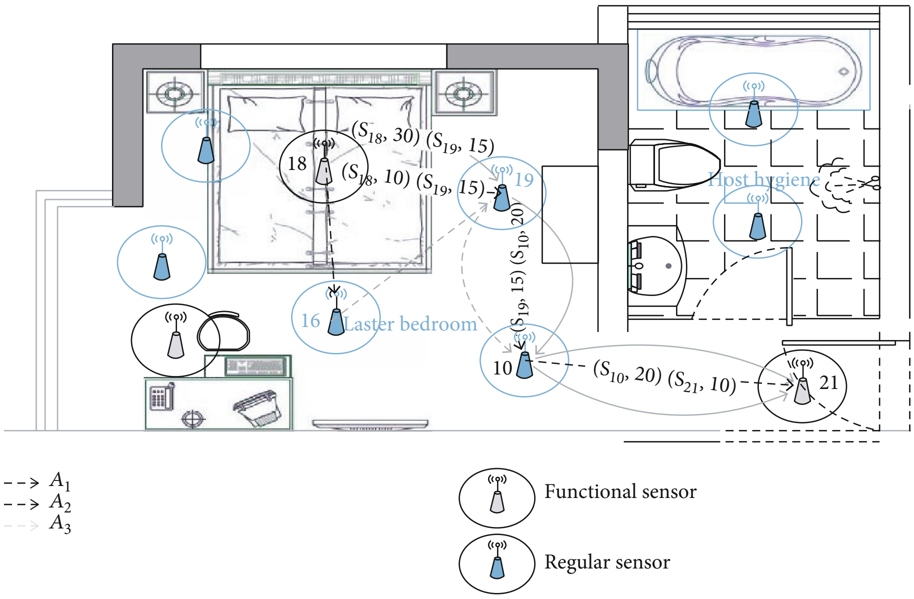

$(S_{18}, 30)$ $(S_{19}, 15)$
18
$(S_{18}, 10)$ $(S_{19}, 15)$
19
$(S_{19}, 15)$ $(S_{10}, 20)$
Host hygiene
16
Laster bedroom
10
$(S_{10}, 20)$ $(S_{21}, 10)$
21
A_1
A_2
A_3
Functional sensor
Regular sensor

Figure 2: Basic layout of sensor deployment in smart environment. Sensors are classified into REGULAR sensors and FUNCTIONAL sensors. Time duration between sensors and sensors' ID is represented by tuples in lines. These lines also represent activity of trajectories.

he/she is/was at particular time; (ii) the long response time and using large bandwidth during computational process are inappropriate for real-time detection. We proposed multilevel privacy controlling scheme (MPCS) to deal with them. (1) *BehaviorPrivacyDeg* is proposed, in order to (i) keep record of user's activity variation and storing these compact patterns into each sensor with the patterns compact algorithm (*PCA*) and (ii) detecting whether the present activity is abnormal or normal based on the abnormal activity detection time-duration algorithm (AADTA). (2) Protecting user's behavior pattern privacy by using multilevel privacy model, server utilizes PLCP to set privacy level according to concern of user. LPA is used to hide the features of user's real location from adversary or untrusted nodes by generating a number of fake locations. The main research contributions of this paper are as follows:

(i) *BehaviorPrivacyDeg*, a novel technique detecting abnormal activity and compact pattern algorithms, is proposed to cache learned parameters using mining training into every sensor node and to sense abnormal activity at real time based on limited resource restrained of sensors

(ii) The multilevel privacy model has been designed to protect users' behavior pattern privacy. Our model not only utilizes PLCP for optimal configuration of privacy levels but also secures user's data from untrusted nodes caused by unpredictable interference in smart IoT environment

(iii) Activity design model, which consists of activity variation, trajectory variation, and duration variation, to define a small difference between two the same activities because the same pattern of activities cannot be repeated exactly in the same way

(iv) Real data-based simulation and experiments have been conducted which showed that our new approach can efficiently protect users' activity and sensitive data in smart IoT environment

The rest of the paper is organized into the following four sections. We thoroughly overview related previous literature in Section 2. We present our new scheme in Section 3. Simulation and experiments are presented in Section 4. Conclusions are discussed in Section 5.

2. Related Work

A number of research studies have been conducted on protecting privacy of users in smart IoT environment. We hereby briefly discuss and compare their findings. Many privacy protection schemes are introduced as means to protect query privacy and users' location privacy for various situations (e.g., snapshot scenario and continuous scenario in navigation apps.). In [8, 10, 11], authors proposed location perturbation, obfuscation techniques, and temporal cloaking techniques, respectively. Generally, all these techniques are deployed to achieve the privacy goal. These proposed techniques can be gained based on trusted third party such as location anonymizes in [12]. In [10, 13, 14], authors have proposed mobile device-based solutions. In some early works, Chow et al. introduced a solution based on location anonymizer to collect the queries of users and forwarding anonymous data set to location-based server (LBS) to protect users' privacy. However, later it is noticed that location anonymizer resulted in the blockage of entire system. In [15], authors proposed two algorithms, named GridDummy and GirDummy that generate dummy location to achieve k -anonymity for user, considering the location's privacy. These two algorithms generated virtual circle and virtual grid

```
Input:$s_l, d_l, c_i, c_l, s_i, d_i$
Output: $r$-activity-patterns, frequent pattern tree (FP-tree) assigned as $f_t$
   While $(d_i, s_i)$ $do$
   if $c_l = REGULAR\ then$ $\text{tree}_{\text{insert}}((s_l, d_l), (d_i, s_i))$ ;
   else if $c_i = REGULAR\ then$ $\text{tree}_{\text{insert}}(s_l, d_l)$ ;
   Tree_insert$((s_l, d_l), ((s_l, d_l))$ ; $else$
   Tree_insert$(s_l, d_l)$ ;
   Tree_insert$((s_l, d_l), (d_i, s_i))$ ;  end
   if $(s_l, d_l) = (d_i, s_i)$; nest item will be assigned in server to $(d_i, s_i)$
   end while
   if $(d_i, s_i)$ last item at end of dataset then
   For every activity $A_l$ in FP-tree do
   if $f_t \geq \lambda\ then$
   add $A_l\ into$ $r$-activity-patterns;
   end if
   end for
   end if
   return $r$-activity-patterns and $f_t$ $(FP - tree)$
```

Algorithm 1: FPMTA.

```
Input: $r$-activity-patterns, $g$-activity-patterns, $\gamma$
Output: $c$-activity-patterns: to compact the real normal activity pattern
   Sorting $g$-activity-patterns in order of descending $|d_{A_u}|$ ; $//|d_{A_u}|$ it represents the quantity of activity in data set $d_A$;
   While $g$-activity-patterns' size = 0 do
   Attain first activity pattern $A_u$ in $g$-activity-patterns
   for activity pattern $A_l \in d_{A_u}\ do$
   Delete $A_l$ in $r$-activity-patterns and $g$-activity-patterns;
   for every activity pattern $A_n$in $g$-activity-patterns do
   Delete $A_l$ in $d_{A_n}$;
   end for
   end for
   delete $A_u$in $g$-activity-patterns;
   sorting $g$-activity-patterns in descending order $|d_{A_u}|$ ;
   end while; $c$ – activity – patterns = $r$ – activity – patterns
   return $c$-activity-patterns
```

Algorithm 2: Patterns Compact Algorithm (PCA).

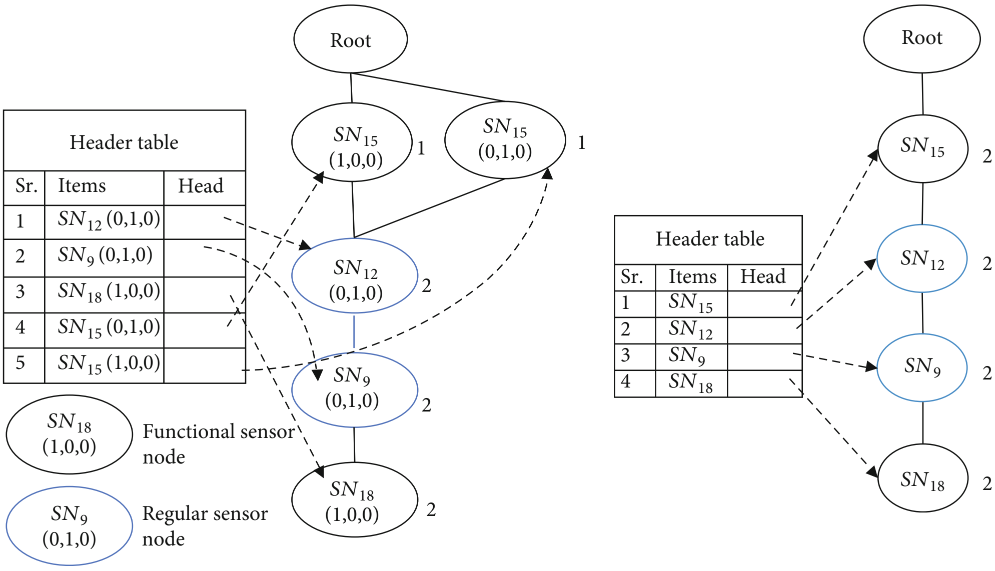
```
Input: table-activity-dect, $d_c$, $t_c$, $D_{A_l}$, $T_{A_l}$, $\gamma$
   $minu = 1$
   $add\ d_c\ into\ T_{A_l}$ ;
   $add\ t_c\ into\ D_{A_l}$ ;
   reorganize $A_l\ with\ D_{A_l}$ and $T_{A_l}$ ;
   for $i \longleftarrow 1$ to table-activity-dect do
   if dissimilar $(A_u, A_l) < minu\ then$
   $minu = dissmilar(A_c, A_l)$ ;
   end if
   end for
   if $minu > \gamma\ then$
   label $A_l$abnormal;
   return c-activity-patterns;
```

Algorithm 3: AADTA.

which were carefully constructed for privacy area of users. However, Lu et al. ignored the background information and query privacy of the users. Although in some recent research studies [16], authors have paid attention to solve the above-mentioned issues thoroughly; however, they introduced heavy system to achieve k-anonymity. In [17], authors proposed a device free localize (DFL) technique which identifies user's location and their activities simultaneously. The wireless signals have the ability to become a sensor itself that can perceive the context information. In near future, this technique may turn the traditional wireless network into intelligent networks. However, the mechanism of this approach is not efficiently working on limited resource-restrained devices. In [18], Liu proposed a scheme for activity recognition using 2D and 3D cameras. However, video-based techniques and approaches can compromise on privacy issues. Moreover, high cost is required for video equipment. In [19], authors have discussed that users' activity in home such as bathing, cooking, and reading can be accessed by

Table 1: Variable detail used in Algorithms 1, 2, and 3.

Variables	Detail
λ	It represents time duration threshold.
γ	It represents variation threshold.
r-activity-patterns	This variable used to store real mined patterns.
g-activity-patterns	It represent the set $\{(A_1, r_{A_u}), \cdots, (A_i, r_{A_u}), \cdots, (A_N, r_{A_N})\}$ and where $A_i \in$ r – act – patterns. And η is number of activities that stored in r-activity-patterns.
r_{A_u}	This variable stores the activities that are the same to A_u.
d_i, t_i, c_i	d_i is for current triggered sensor device/node, and t_i represents the time duration probability. c_i is category of sensor device d_i (REGULAR sensor device or FUNCTIONAL sensor device).
d_l, t_l, c_l	d_l represents last triggered sensor, and t_l represents corresponding time duration probability. c_l is for corresponding type/category.
c-activity-patterns	This variable is used to store the compacted patterns from the real.
A_c, d_c, t_c	A_c represents current user' activity, d_c represents current sensor device, t_c is current time duration probability of sensor device.
D_{A_l}, T_{A_l}	These represent their mining from latest change of state-message.



Header table

Sr.	Items	Head
1	SN_{12} (0,1,0)	
2	SN_9 (0,1,0)	
3	SN_{18} (1,0,0)	
4	SN_{15} (0,1,0)	
5	SN_{15} (1,0,0)	

Header table

Sr.	Items	Head
1	SN_{15}	
2	SN_{12}	
3	SN_9	
4	SN_{18}	

Figure 3: Basic difference between two kinds of frequent pattern tree (FP-tree). (a) The trajectory FP-tree and (b) the time duration-based FP-tree.

Table 2: Detects abnormal activity table of sensor device S_{16}.

Previous device (d_l)	Previous time-duration probability (t_l)	Particular time-duration probability (t_o)
S_{20}	(1,0,0)	(0.8, 0.2, 0)
S_{15}, S_{11}, S_{12}	(1,0,0), (0,1,0), (0,1,0)	(1,0,0)

unauthorized entities on the wireless network, even all communications are encrypted. In this approach, authors used fingerprints and time-based snooping (FATS) attacks. However, chances of privacy leakage of users' activity are very high due to limitation of this approach in [20–22]; temporal cloaking and spatial assessed time-location are directed to the main server instead of the accurate value. The main focus of these approaches is to prevent exact identification of user's location and thus improving privacy. These techniques harm the timeliness and accuracy of the responses from server, and more seriously, there are some upfront attacks that could still break user privacy. In [23], authors have proposed k-pattern clustering algorithm that classifies complex and varied user activities. This approach also used Allen's temporal relation to predict and recognize users' activities inside home. However, this method did not focus on privacy of users' activities as well as location-based privacy of users. If we observe carefully, most of the recent techniques have some pitfalls such as usage or trust on the third party or server and time-consuming huge processing overhead. In [24], authors provide new system for security institutes to monitor abnormal events. With the help of deep

Table 3: Several levels for protecting privacy of users.

Privacy level	Type	Description
PL1	Zone/region	In this level, user just shares that he is in university but does not allow to share where exactly he is (e.g., in which building and apartment).
PL2	Building/office/room	In this level, user shares his room/office/building but does not share the exact time (e.g., in which office at what time he is/was).
PL3	Time duration	In this level, either he shares approximate time (09.00 to 14.00) about his activity/location or accurate time, day, week, and month etc.
PL4	Activity/action	This level includes the activity/action of the user (e.g., exercise, work, taking class, and watching TV).

learning, authors attained high performance of human behavior recognition by using model tests and training but his scheme does not enable user to define privacy level according to user wish. In [25], authors proposed novel idea based on genetic algorithm to resolve classification problems based on sensor data but they also ignore privacy of user based on sensor data.

Our proposed scheme is different from traditional approaches because our research emphasizes on the user's behavior pattern privacy, including behavior pattern privacy degree, multilevel privacy model, location protection mechanism, and detection algorithm.

3. Multilevel Privacy Controlling Scheme (MPCS)

In this section, we present behavior pattern privacy degree (*BehaviorPrivacyDeg)* and multilevel privacy model of proposed multilevel privacy controlling scheme in detail.

3.1. Behavior Pattern Privacy Degree. Behavior pattern privacy degree (*BehaviorPrivacyDeg*) is aimed at protecting privacy of user's activity variation in smart IoT environment which is as follows: (i) first, it extracts normal behavior pattern from the genuine data and then presents an activity pattern algorithm based on time duration that compresses and reduces the quantity of mined behavior pattern of user's activity; (ii) secondly, it records mined pattern in each device according to record keeping mechanism, and it also detects abnormal activity to protect user's behavior and pattern privacy. *BehaviorPrivacyDeg* uses three algorithms to protect the privacy of user's behavior pattern which are (i) frequent pattern mining-based time-duration algorithm (FPMTA), (ii) patterns compact algorithm (PCA), and (iii) abnormal activity detection time-duration algorithm (AADTA). *Sensors*: we divided sensors into *REGULAR* sensors and *FUNCTIONAL* sensors as per requirement of deployment to sense the data of user's locations and activities as shown in Figure 2. Firstly, set of all the deployed motion sensor devices across the smart IoT environment is represented as $D = \{s_1, s_2, s_3, s_l, \cdots, s_n\}$. User's position is represented by sing location of sensor device s_l which detects the movement of user's position/location P. Sensor devices are defined by N. As we know, all users probably have different velocity of doing activities. Therefore, the time between these sensors during user's activity is different and longer as compared to specific time segment. Activity is produced that is composed of atomic users' activities. Atomic activities $a_{l_a}^i = (s_{l_a}, t_{l_a}^i)$ define the trigger of sensor device s_l where $s_{l_a} \in D$, and $t_{l_a}^i$ is trigger time of s_{l_a} in ith sampling period. Number of sampling periods is defined by N_s which represents the conditions when a person passes by s_{l_a}. Basic activity is defined as $\beta_{l_a} = \{\alpha_{l_a}^1, \alpha_{l_a}^2, \alpha_{l_a}^3, \cdots, \alpha_{l_a}^{N_s}\} = \{(s_{l_a}, t_{l_a}^1), \cdots, (s_{i_a}, t_{l_a}^{N_s})\}$ which shows basic activity, where time duration is $d_{l_a} = t_{l_a}^{N_s}, t_{l_a}^1$.

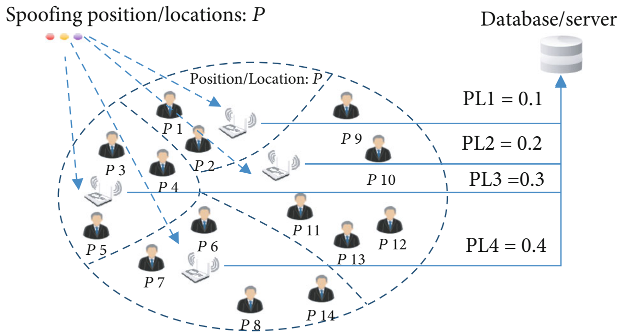


Figure 4: All privacy level that queries to the server. Work flow of PLCP.

3.2. Frequent Pattern Storage, Compression, and Mining. To store, compress, and detect the abnormal activity, top priority of *BehaviorPrivacyDeg* is mining the user's normal activity pattern to protect behavior pattern.

Definition 1. Normal activity is defined as if frequency of an activity A_u which we assigned as f_u exceeds a particular threshold during appearing in the storage data; then, activity A_u is called a normal activity.

Definition 2. Abnormal activity can be defined as activity that deviates from normal activities in the collected data. In activity recognition, the temporal relationship is foundation of sequence determination [26], and it leads to error of activity recognition. We determine abnormal activity as follows, if there is any kind of activity pattern A_u which apparently seems normal but actually has deviation from normal activity, i.e., $A_{\text{vary}} \leq \gamma, Au$, is determined as abnormal activity. Mostly, supervised learning algorithms for sensor data require several labeled data; therefore, learning algorithms

```
Input. $\rho$ : the desired level of privacy, $\{\rho_i\}$: the level of privacy for user in the data, $\{(\partial_j, \delta_j)\}$: the value for the data model, $s_{ij}$: learned data
$\mathrm{PL}_{\rho_i} \longleftarrow \sum_{j=1}^{k} \partial_j \Pr(\rho_i, \partial_j, \delta_j)$
$T \longleftarrow$ search the set of $\rho_i$, so that $|\rho - \rho_i| < \epsilon$;
$C_{\mathrm{opt}} \longleftarrow \sum_{j=1}^{k} \partial_j$;
Optconf $\longleftarrow \varnothing$ ;
for every user $u_i$ having $\rho_i \in T$ do
    $\mathrm{PL1} \longleftarrow \sum_{j=1}^{k} \partial_j s_{ij}$ ;
    if $C_{\mathrm{opt}} > |\rho_i - \mathrm{PL1}| < \epsilon$then
        $C_{\mathrm{opt}} > |\rho_i - \mathrm{PL1}|$;
        Optconf $\longleftarrow u_i{}'s$ privacy configuration;
    end if
Return Optconf
```

Algorithm 4: Level concern privacy algorithm (LCPA).

unsupervised that saved labor and accelerate the learning speed [27].

3.3. Frequent Pattern Mining. Keeping in mind the Definition 1, we prefer to use frequent pattern mining approach [7] for user' behavior pattern privacy by mining normal activity patterns. Based on frequent pattern mining approach [7], if frequency of an element set exceeds minimum threshold λ within specific time duration, then it is considered as a normal activity. Each path from leaf node to root node and root node to leaf node is defined as pattern A_p, and the frequency is calculated as f_p which represents minimum support count in a path. We use frequent pattern tree (FP-tree) to store quantitative and crucial information about FP-tree and time duration. FP-tree is proposed to achieve the privacy level of data in smart IoT environment. In FPMTA, line 3 and line 5 represent the *insert-tree* function. The function of insert-tree set $((s_i, d_i), (s_l, d_l))$ is inserted in two steps; in first step, node (s_i, d_i) is inserted into *FP-tree* as a child node (s_l, d_l), and in second step, (s_i, d_i) *insert-tree* is to insert a node (s_i, d_i) into *FP-tree* as a child node of root node. If there (s_l, d_l) is a child node (s_p, d_p) or root node which $|s_p - d_i| \leq \lambda$ and $s_p = s_i$, then counting of s_p, d_p is incremental by value 1. Suppose it is not the same, then node (s_i, d_i) is inserted into *FP-tree* as fresh child node's root node (s_l, d_l). Variables of Algorithms 1, 2, and 3 are used in Table 1.

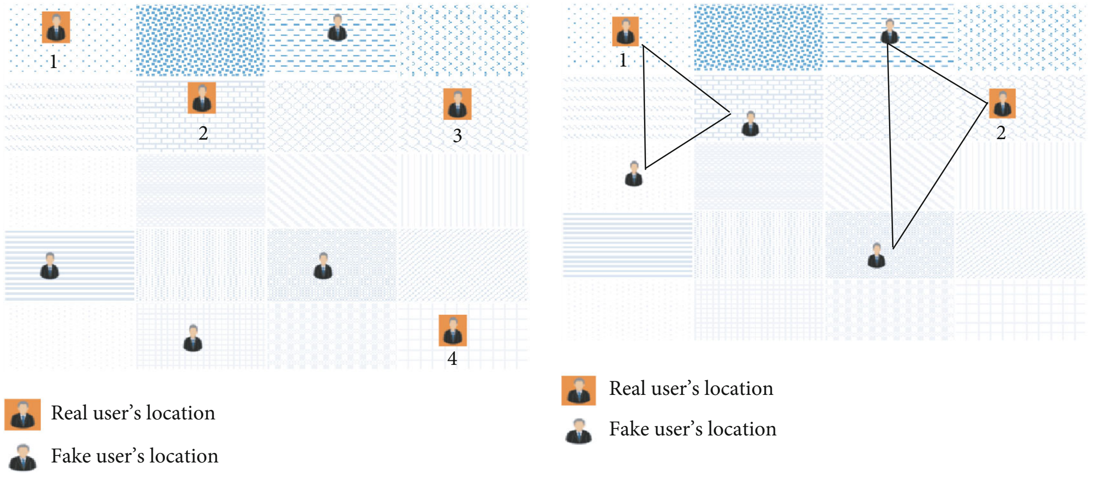
To compress and compact the mined frequent activity pattern of user's behavior, we introduced a PCA. Furthermore, *BehaviorPrivacyDeg* of MPCS introduced abnormal activity detection-based time-duration algorithm (AADTA) to protect the privacy of user's behavior pattern by detecting abnormal activity. AADTA contains sensor device ID d_i and sensor category and table of activity detection that is named as table-activity-dect. Activity table of sensor device S_{16} as shown in Figure 3 is described in Table 2. Mined patterns are stored in relevant room sensor devices separately as per proposed storage method. *Previous sensors* stored the ID in normal pattern field before triggering sensor S_{16}. Time-duration probability is stored by previous time-duration probability corresponding with previous sensors S_{16}.

3.4. Multilevel Privacy Design Model. The term privacy conveys various concepts such as privacy of activities, location, time duration, and decisional privacy. The form of privacy discussed in this section is user's behavior pattern privacy based on activities. We divided user's behavior pattern privacy into four levels termed as privacy level-1 (PL1), privacy level-2 (PL2), and so on as discussed in Table 3 and Figure 4. Let $\mathrm{PM} = \{\mathrm{PL}_1, \mathrm{PL}_2, \mathrm{PL}_3, \mathrm{PL}_4\}$ be the set of privacy model, including four privacy levels. The ability of multilevel privacy model is to deal privacy of user's behavior pattern in smart IoT environment. Multilevel privacy model is comprised of (i) privacy level configuration protocol (PLCP) and (ii) activity design model.

3.5. Privacy-Level Configuration Protocol (PLCP). PLCP is designed to manage privacy of users by controlling privacy levels and transmit data among sensors. In order to avail any service from server, users have to share some information of their privacy level with the server through limited resource sensors as shown in Figure 2. Privacy of user will be changed with the selection of privacy level. Term u_i is for user, and term ρ_i is used for privacy-level concern. At the level PL_i, the average number of hidden data for all user is defined as $\mathrm{PL}_{\rho_i} = \sum_{j=1}^{k} \partial_j \Pr(\rho_i, \partial_j, \delta_j)$ where term δ_j is used as how sensitive the data is perceived by user and ∂_j is used as weight for the data. The $\Pr = (\rho_i, \partial_j, \delta_j)$ bt the value of user's privacy concern. We defined this measure for privacy rating at privacy configuration level PL_{ρ_i}. For the user u_i, the actual weighted number of hidden data $\sum_{j=1}^{k} \partial_j s_{ij}$ is privacy rating at level of PL_{ρ_i}. PLCP uses level concern algorithm and privacy level index mechanism.

$$\sum_{j=1}^{k} \partial_j s_{ij} = \sum_{j=1}^{k} \partial_j \Pr\left(\rho_i, \partial_j, \delta_j\right). \qquad (1)$$

(1) *Level Concern Privacy Algorithm (LCPA).* LCPA provides a way for finding the optimal privacy



(a) Real and fake users' location

(b) Fake location with bigger and smaller cloaking region (CR)

Figure 5: Layout of our basic approach in smart environment.

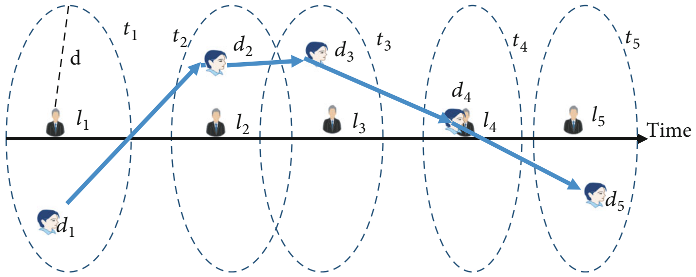


Figure 6: The scenario of user's locations inside smart environment.

configuration for a desired level of privacy concern. A new user can stipulate his/her level of privacy concern ρ_i based on the relative value in the range (1 to 4). LCPA assumes the data item models $\Pr = (\rho_i, \partial_j, \delta_j), j = 1 \cdots ..k$. Also, privacy configuration for each user has been calculated as ρ_i, $i = 1,2,3,4$. LCPA first calculates the privacy level PL_{ρ_i} for user level of privacy concern ρ_i with the data item model and then searches whether the user has the similar level of privacy concern $|\rho - \rho_i| < \in$, where $\in > 0$ is very small value according to LCPA.

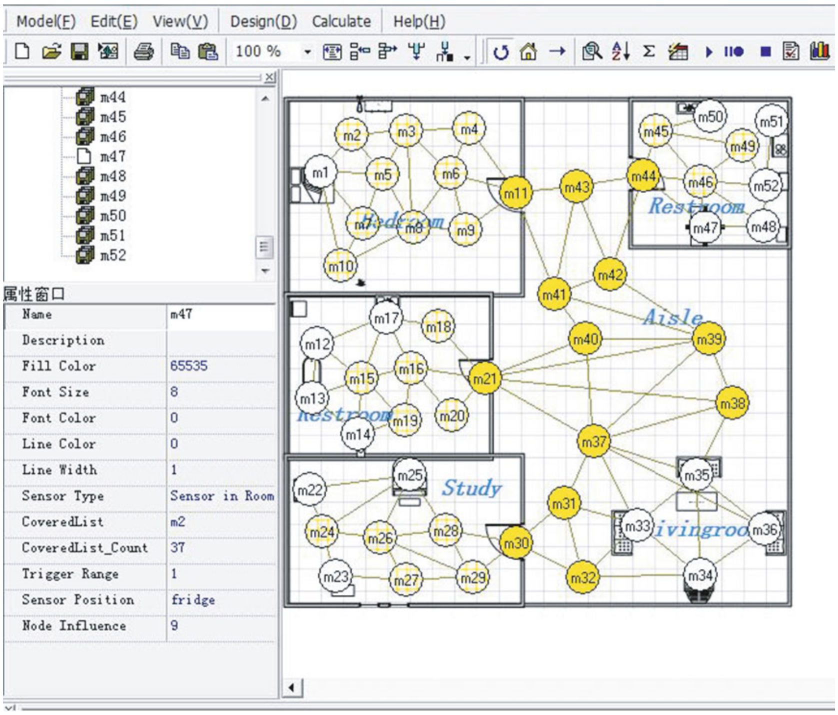
(2) *Privacy Level Index Mechanism*. We introduced a new privacy level index mechanism which is used to assign index for each level. Let us assume that privacy level has been PL1-assigned index 0.1, PL2 has been assigned 0.2, PL3 has been assigned 0.3, and PL4 has been assigned index 0.4. User can use these indices to set their privacy level according to their concern in our smart IoT environment. At the same time, different user has different privacy levels and these levels are used by our proposed MPCS to protect the user's behavior pattern privacy.

Each privacy level has data set as discussed above. Sometimes the user is more conscious about information of his location and sometimes about information of his time duration etc. Figure 4 illustrates how PLCP works. Assuming that one user follows the PL1 in Figure 4 when sensor is triggered, it first executes the LCPA to control and manage privacy of user's behavior pattern. The target of adversary is to access sensitive information of a user. We focused on two types of adversaries: (i) active adversary, any entity is an active adversary if he can access the untrusted sensor nodes. (ii) A passive adversary, which can eavesdrop on a communication channel between compromised nodes to track other user's sensitive data. We consider gateway and sensor nodes as active adversaries.

3.6. Location-Based Privacy Algorithm (LPA). Privacy levels 1 and 2 include user's location, and in order to protect user's location, we used concept of entropy. Entropy is used to measure the degree of k-anonymity. To calculate entropy, each location has probability of being queried q_i and probability donated by b_i is 1. To identify the individual's entropy E in users, set is defined as

$$E = -\sum_{i=1}^{k} b_i \cdot \log 2b_i. \tag{2}$$

Our goal is to attain the maximum entropy, which can be achieved when all possible positions/locations P have the same probability $1/P$ where the maximum entropy will be $E_{\max} = \log_2 P$. Server can assume real location with high probability as $1/P - P_d$, where P_d represents the number of



属性窗口

Name	m47
Description	
Fill Color	65535
Font Size	8
Font Color	0
Line Color	0
Line Width	1
Sensor Type	Sensor in Room
CoveredList	m2
CoveredList_Count	37
Trigger Range	1
Sensor Position	fridge
Node Influence	9

Figure 7: Motion sensor-based smart environment setup for simulation experiment.

fake locations and server will calculate it based on probabilities of their low query. The query probability is higher than others in locations 1 to 3 and on the basis of information.

It means that $P - P_d = 3$,and entropy will drop considerably from $\log_2 P$ to $\log_2 (P - P_d)$. We enhance privacy of users in two phases: (i) first, we try to select fake locations of users with the same query probabilities; (ii) second, if there are more than one user, the fake location spread is as far as possible. Suppose the user's location map is segregated into equal size cells $n \times n$ as shown in Figure 5. Each cell of the map has its own enquiry probability that is based on previous query history as follows:

$$q_i = \frac{\text{number of enqueries in each cell } i}{\text{number of enqueries in full map}}, i = 1, 2, 3, \cdots \cdots \cdots .n^2, \tag{3}$$

where

$$\sum_{i=1}^{n^2} q_i = 1. \tag{4}$$

To provide a degree of k-anonymity, in addition to real locations, we need to conclude the other $k - 1$ cells to assign the fake locations. The user selects the P cells right before and P cells right after real location from sorted list as $2P$ users. Therefore, user make N set of cells, and in every set, one cell belongs to real user's location and the others are randomly selected from $2P$ users. The $m^{\text{th}} (m \in [1, n)$ set is represented as $R_j = [r_{j1}, r_{j2}, \cdots, r_{ji}, \cdots, r_{jk}]$. The normalized query probability of the involved cells which is based on real query probabilities of the selected cells can be presented as s_{j1}, $s_{j2}, \cdots, s_{ji}, \cdots, s_{jk}$ and calculated by summing it to 1.

$$s_{j1} = \frac{q_{bi}}{\sum_{l=1}^{k} q_{bl}}, i = 1, 2, \cdots \cdots , k. \tag{5}$$

To effectively achieve k-anonymity of P location, we need to create an optimal set. The level of privacy is guaranteed by using the entropy metric that is extensively used to measure privacy of users. We compute entropy for specifically selected set R_j as follows:

$$E_j = -\sum_{i=1}^{k} r_{ji.} \cdot \log 2\, r_{ji.} \tag{6}$$

Finally, the LPA achieved the set with effective and highest degree of entropy.

$$R = \operatorname{argmax} E_j. \tag{7}$$

To measure the cloak region (CR), distances between pair of fake locations are calculated and the sum of distances can be utilized to measure the CR which is $\sum_{i=j} d\,(r_i, r_j)$ where $d\,(r_i, r_j)$ represents the distance between rows/cells r_i and r_j. In Figure 6, l_1 represents real location of user and d_1 is selected as a fake location of the user, since it is considered farthest location from l_1. Furthermore, suppose there are two choices for assigning third fake locations d_2 and d_3. We select it based on the sum of distance between

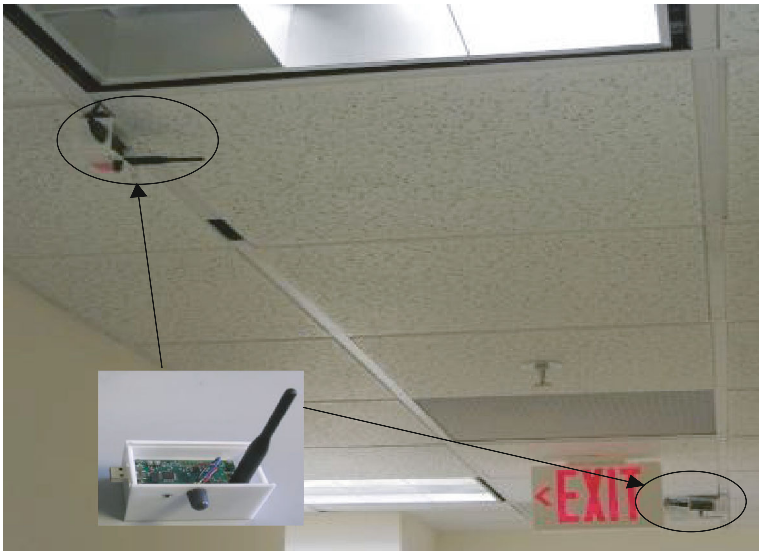


Figure 8: Example of the smart IoT environment and deployment of motion sensor devices are shown.

Table 4: Experiment result of performance test.

Number of pattern	Average time of execution
6	77.5 m/s
12	80.4 m/s
18	86.3 m/s
24	93.9 m/s

Table 5: Detecting abnormalities.

Parameters	Algorithm detecting_ activity trajectory	
Size of model	95	95
Trajectory_abnormality	73	73
Disc_abnormal_trajectory	75	91
Size of model	54	54
Time_duration_abnormality	38	38
Disc_time_duration_abnormality	35	Fail

pairs of fake user's locations. We have to select either of them because $d_2 l_1 + d_2 d_1 = d_3 l_1 + d_3 d_1$. In this scenario, $d_2 l_1 . d_2 d_1 = d_3 l_1 . d_3 d_1$; hence, we select d_2 as a fake location. Let $R = [r_1, r_2, \cdots, r_k]$ represents the set of fake and real user's location. Multiobjective optimization problem (MOP) is described as

$$\max \left\{ -\sum_{i=1}^{k} b_i \cdot \log_2 b_i, \prod_{i \neq j} d\left(r_i, r_j\right) \right\}, \tag{8}$$

where $r_i, r_j \in R$, k_i, and k_j represent the query probabilities of the r_i and r_j, respectively. Our first priority is to confuse the adversary so that adversary cannot target the specific location of user. This objective can be represented as follows:

$$R = \operatorname{argmax}\left(-\sum_{i=1}^{k} b_i \cdot \log_2 b_i \right), \tag{9}$$

That is basic condition to achieve the higher entropy by using a set of fake locations. Optimal combination of P locations is as follows:

$$R = \operatorname{argmax} \prod_{i \neq j} d\left(r_i, r_j\right). \tag{10}$$

Time Duration. The time duration T_d is divided into three parts: small, medium, and big; thus, fuzzy logic [28] is used to calculate the time duration, and fuzzy inference system (FIS) [29] is adopted to measure the probability of T_d being small $(P_{ş-T_d})$, medium $(P_{ɱ-T_d})$, and big $(P_{ƀ-T_d})$. Basic activity is defined as β_{l_a}, and sensor device is defined as s_{di} so as a result $\beta = (T_{d_a}, s_{i_a})$ is redefined as $\beta = (s_{i_a}, (P_{ş-T_d}, P_{ɱ-T_d}, P_{ƀ-T_d})$. In this paper, small time duration range is 0 to $\tilde{d}_{t3}$, and medium time duration range is from $\tilde{d}_{t1}$ to $\tilde{d}_{t4}$, and big time duration range is from $\tilde{d}_{t2}$, where $0 < \tilde{d}_{t1} < \tilde{d}_{t2} < \tilde{d}_{t3} < \tilde{d}_{t4}$. Each s_{i_a} stores $t\_\text{rule}^i = \{d^i_{t1}, \tilde{d}^i_{t2}, \tilde{d}^i_{t3}, \tilde{d}^i_{t4}\}$, and t_rule^i is fixed according to location and monitoring zone of sensor device s_{id}. The mean of maximum scheme is appropriate for our method. Assumed activity A_1 as an example and we set $\tilde{d}_{t1} = 5$, $\tilde{d}_{t2} = 20$, $\tilde{d}_{t3} = 35$, and $\tilde{d}_{t4} = 50$. After using fuzzy logic, term $(s_8, 15)$ can be defined as $(s_8, (0, 1, 0))$.

Activity Design Model. In this section, we described the concept of activity variation. Activity variation can be defined as small difference between two the same activities because the same pattern of activities cannot be repeated exactly in the same way. Activity variation consists of trajectory variation and duration variation which is used to measure this small difference.

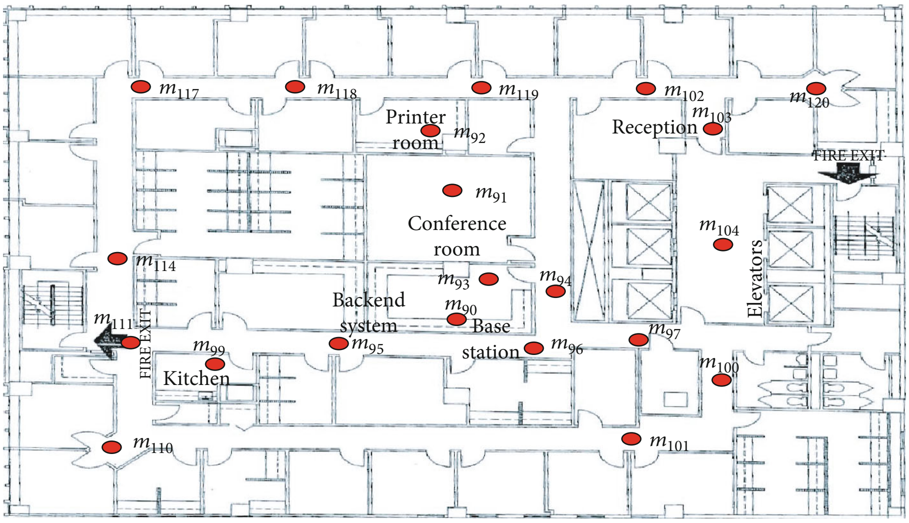
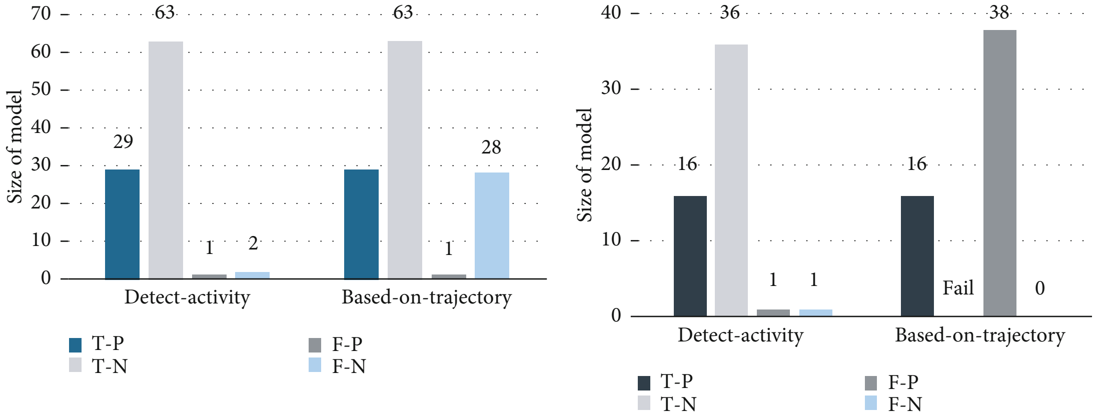


FIGURE 9: Deployment layout of motion sensor devices network in smart IoT environment. The position of sensor and its ID are shown.



(a) Detection abnormal trajectory

(b) Detection abnormal time duration

FIGURE 10: Experiment results during simulation test.

TABLE 6: Detail of stored patterns in sensors.

Sensor device ID	Total patterns	Sensor device ID	Total patterns	Sensor device ID	Total patterns
d_1	2	d_{13}	17	d_{25}	17
d_2	4	d_{14}	14	d_{26}	19
d_3	7	d_{15}	13	d_{27}	14
d_4	9	d_{16}	9	d_{28}	13
d_5	12	d_{17}	12	d_{29}	13
d_6	15	d_{18}	7	d_{30}	11
d_7	17	d_{19}	8	d_{31}	9
d_8	13	d_{20}	10	d_{32}	7
d_9	15	d_{21}	13	d_{33}	5
d_{10}	16	d_{22}	11	d_{34}	3
d_{11}	11	d_{23}	12	d_{35}	2
d_{12}	15	d_{24}	15	d_{36}	7

(a) *Trajectory Variation*. The term trajectory variation is defined as T_v. Activities A_0 and A_1 as shown in Figure 5 take as an example $T_{\mathrm{va1}} = s_{18} \longrightarrow s_{20} \longrightarrow s_{20} \longrightarrow s_{21}$ but $T_{\mathrm{va2}} = s_{18} \longrightarrow s_{10} \longrightarrow s_{20} \longrightarrow s_{21}$, and this represents the same activity but with a small difference in trajectory. This trajectory variation is measured by M_variation, and the difference between two trajectories T_{vn} and T_{vm} is calculated as

$$T_{\mathrm{variation}}\left(T_{vn}, T_{vm}\right) = \mathrm{Minu}\left(\left|T_{vn} - T_{vm}\right|, \left|T_{vm} - T_{vn}\right|\right) + \left\|T_{vn}\right| - \left|T_{vm}\right\| + \mathrm{order}\left(T_{vn}, T_{vm}\right) = \mathrm{M}_{\mathrm{variation}}. \tag{11}$$

$|T_{vn} - T_{vm}|$ represents the total number of s_{di} which $s_{di} \in T_{vn}$ and $s_{di} \notin T_{vm}$. $\|T_{vn}| - |T_{vm}\|$ explain the length between T_{vn} and T_{vm}. Order (T_{vn}, T_{vm}) computes the difference in sequence between T_{vn} and T_{vm} [26].

```
Input: real location $L_{\text{real}}$, sets of $N$ and $P$, probabilities of query in $q_i$.
Output: set of fake-locations
    All cells sort on based probabilities of their query
    Select fake $2P$ of users among which $P$ user is right before $L_{\text{real}}$ and $P$ user right after $L_{\text{real}}$in stored list.
    for $(m = 1; j \leq N; m++)$ do
    develop a set $R_j$ which consist of $L_{\text{real}}$and $P-1$, additional cells are randomly chosen from users $2P$;
    Calculates the normalized probability $s_{ji}$ for every cell $r_{ji}$ in the set.
    $E_j \longleftarrow -\sum_{i=1}^{k} b_{ji} \cdot \log_2 b_{ji}$ ;
    End
    Output max $E_j$;
```

Algorithm 5: Location-based privacy algorithm

(b) *Time Duration Variation*. As discussed above, activities A_0 and A_1 as shown in Figure 5 are not the same activities due to the difference of time duration in s_{19}. However, another activity $A_y = \{(s_{10}, 11), (s_{18}, 25), (s_{19}, 13), (s_{18}, 15)\}$ is not same with A_1, and difference of time duration is small. Term T_{dv} is used for time duration variation. Therefore, the variation between the duration of two activities can be calculated as

$$\begin{aligned} \text{Dissimilarity}\left(A_x, A_y\right) &= \frac{T_{dv} - \text{deviation}\left(T_{vx}, T_{vy}, D_{A_x}, D_{A_y}\right)}{(|T_{vn}, T_{vm}|)/2} \\ &+ \frac{T_v - \text{deviation}\left(A_x, A_y\right)}{(|T_{vn}, T_{vm}|)/2} \\ &= A_{\text{dissmilarity.}} \end{aligned} \quad (12)$$

Activity variation of PL4 is calculated by equations (10) and (12), where η is the duration threshold. The variation threshold is defined as Γ to measure the similarity, and if $A_{\text{simlarity}} \leq \Gamma$, then A_x is considered as similar to A_y.

4. Experiments

4.1. Simulation-Based Experiment. As a simulation model with ground facts, we used smart IoT environment simulator tool to simulate the sensor device-based smart IoT environment, and information was installed manually instead of real setup smart IoT environment. Simulation smart IoT environment is basically divided into three main parts which are as follows.

(1) *Motions Sensor Devices*. We installed more than 100 sensor devices to sense data of location-based users' activity for simulation in smart IoT environment which is shown in Figures 7 and 8. In Figure 7, sensors, which are colored with yellow, are deployed in hallways and elevators. Light yellow sensors are deployed within the rooms, office, and conference rooms. White color sensors are installed in living room, study room, and restrooms.

Table 7: Triggered sensors (known versus unknown).

Task setting	Known	Unknown	ADL
105-115	9	11	73%
115-106	12	15	77%
104-121	10	11	80.10%
112A-109B	4	6	62%
101-119	6	8	72%
Average ADL			73%

(2) *Smart IoT Environment's Trajectory*. We designed more than 15 normal trajectories which have average length of 13. These trajectories reflect typical condition about user's activities.

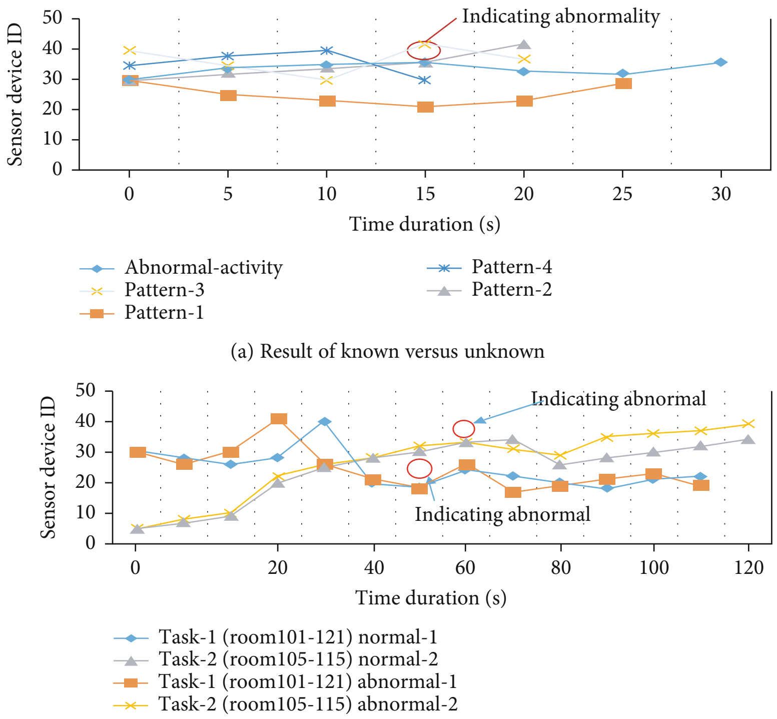
(3) *Time Duration*. As per deployment locations of sensor devices (d_i's locations) and basic features, three types of t_rule are defined to respond the concerned sensor devices.

(i) In t_rule1, firstly, {2 d, 4 d, 6 d, 8 d} is designed for those sensor devices which are utilized for detecting passing (such as in lobby and hallway).

(ii) In t_rule2, {1 s, 3 s, 5 s, 9 s, 11 s, 13 s} is designed for such sensor devices which are deployed in areas where users may stay for few minutes (such as in washroom and kitchen).

(iii) In t_rule3, {0.4 h, 1.5 h, 2 h, 5 h, 7 h, 9 h} is designed for sensor devices which are located in the area where users will stay for rather long time such as office, study room, and bedroom.

Meanwhile, their time duration of staying is t_d^i, and corresponding *table-activity-dect* are set and assigned with appropriate value manually. The simulation detection system has completed the operations of the LPA, FPMTA, PCA, and the AADTA.



(a) Result of known versus unknown

(b) Result of task normal versus abnormal

Figure 11: Experimental result of tasks.

Real-Time Location. The parameter average distance and location are designed to calculate the real-time location' property. ADL is measured as follows:

$$\mathrm{ADT} = \sum_{i=1}^{n} \frac{L_{\mathrm{dis}}}{|A_u|}. \quad (13)$$

L_{dis} represents the trigger sensor devices during decision-making, and A_u represents the length of A_u. Experiment results showed that ADT of detecting activity is 75.5% which is good as compared to centralized detecting algorithm.

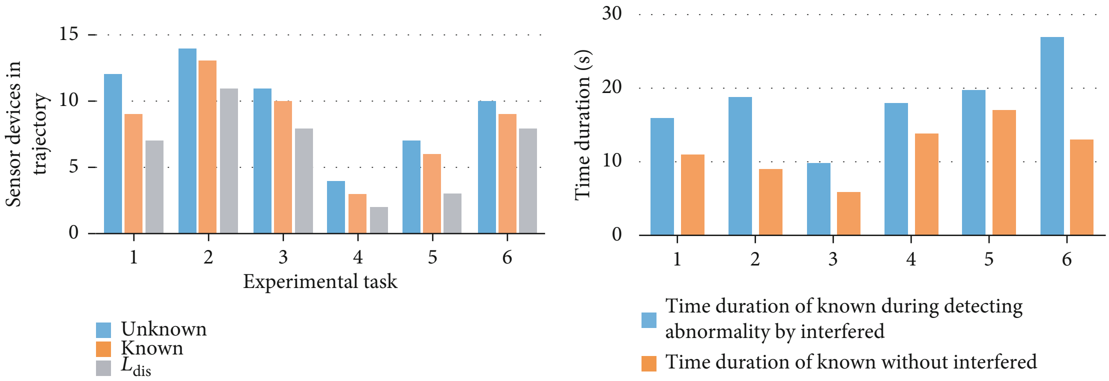
4.2. Lab-Time Experiments. In this section, we conducted real experiment.

4.2.1. Detecting Activity's Feasibility. In smart IoT environment, each sensor device will use AADTA for execution process. In AADTA, the time complexity is $o(t^i)$ and it showed that the time complexity of AADTA is $o(t^i)$. We used TelosW sensor devices for real-time experiment because TelosW has memory size of 1 MB, and it meets the computing capacity of detecting activity. If the average size of stored patterns is 10 at TelosW sensor device, then it means total 7489 patterns $((\text{bytes})(1024 * 1024)/(\text{bytes})((4+4+4+2) * 10) = 7489)$ can be stored on one sensor device in smart IoT environment. It clearly showed that feasibility of sensors' capacity for storage of patterns is enough. Average time of execution of number of patterns is shown in Table 4.

Detecting Abnormalities. Transition probabilities of each sensor in smart IoT environment are represented by $T\_\mathrm{pro}^i = \{t\_p\_r^i, t\_p\_u^i, t\_p\_l^i, t\_p\_d^i\}$. This transaction probability is set to calculate the possibility of which near sensor device will be triggered for next. Considering the deployed sensor devices as shown in Figure 7, if a user triggers m_{27}, the user must trigger $\{m_{26}, m_{22}, m_{25}, m_{24}\}$ as transaction $m_{27} \longrightarrow m_{26}, m_{27} \longrightarrow m_{22}, m_{27} \longrightarrow m_{25}, m_{27} \longrightarrow m_{24}$. If we set $T\_\mathrm{pro}^i = \{0.2, 0.1, 0.3, 0.4\}$ then user will like to select the trajectory $m_{27} \longrightarrow m_{25}$. Moreover, it is also possible that user may choose to do the remaining three trajectories. Here, users are allowed to randomly choose any trajectory from 15 designed trajectories. In other words, users can choose any route depending on the T_pro^i and user can also change his route. We calculated 95 trajectories after repeating 95 times, and only 4 of them are the same as we have designed. 75 abnormal trajectories are detected by our algorithm-based trajectory method [7] and labeled 91 abnormalities, but in real, just 73 abnormalities are produced as shown in Table 5. We use two important keys during simulation experiment when time duration is taking into consideration. Firstly, we use a^i for average speed where i represents the sensor device ID. Average speed represents the approach corresponding with every interlinked device-pair but we set up various speeds during simulation in each sensor device to manage the average speed. Secondly, we assign various speeds with index v^i representing the variance of a^i. When user is passing through sensor device d_i and a^i randomly selects from 0.4 m/s to 1.2 m/s, v^i randomly selects from 0.11 m/s to 0.32 m/s. Time duration t_d^i is altered manually. After repeating and executing 40 times, 40 trajectories are produced with uniform time duration. 30 abnormalities are generated, and our algorithm detected 29 abnormalities by using trajectory-based approach [7].

TABLE 8: Time-duration number of devices triggered in normal versus abnormal.

Task setting	Senor devices				Time duration (s)	Abnormal-detection-location	ADL
105-115	9	9	48.51	69.1	7	58%	
115-106	12	12	80.2	85.43	8	68.4%	
104-121	10	10	52.31	65.89	7	78.9%	
112A-109B	4	4	12.10	24.23	4	70%	
101-119	6	6	16.75	25.13	3	80%	
					Average ADL	71%	



(a) Experiment of known versus unknown

(b) Time duration during interfered versus normal

FIGURE 12: Experiment results of our proposed scheme.

4.2.2. Results. The experimental setup to validate our algorithms is based at Chongqing University Campus A, China. During these experiments, we choose two groups of students who have volunteered to participate. Students in group 1 were aware with the environment layout, and students of group 2 were not familiar with environment. Sensor devices were deployed in the building as shown in Figure 9. Red colored sensors in Figure 9 represent the motion sensors. TelosW sensor devices were deployed, and position of sensor in building is shown in Figure 8. Five tasks were performed in two experiments. In each task, participant needs to start from specific position and reaches destination through designed workplace. To achieve the fair result, the specified rooms and position were randomly chosen. The results are shown in Figure 10(b). After extracting 662 activates, we stored related information in each node by LCPA, LPA, PCA, and AADTA, and Table 6 shows the complete details.

Knowing and Unknowing. Students of group 1 were aware about the layout of designed setup, and they completed all six tasks without any prompting. Trajectories of group 1 are traced to detect abnormal activity at real time by using Algorithm 5 (AADTA). Students of group 1 involved in the same task are different from unaware participants of group 2 as shown in Table 7. In other words, unaware participants develop uncommon trajectories which were significantly different from pattern generated by aware group. After 14.5 seconds, it is clearly shown that it repels previous possibility of pattern 2 and it mismatches with other patterns shown in Figure 11(a). Therefore, such kind of activity is labeled with abnormal activity, and user's behavior pattern privacy can be protected by detecting such abnormal activity.

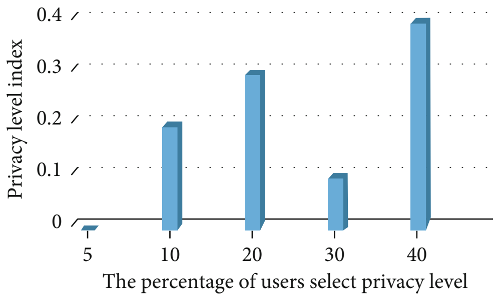
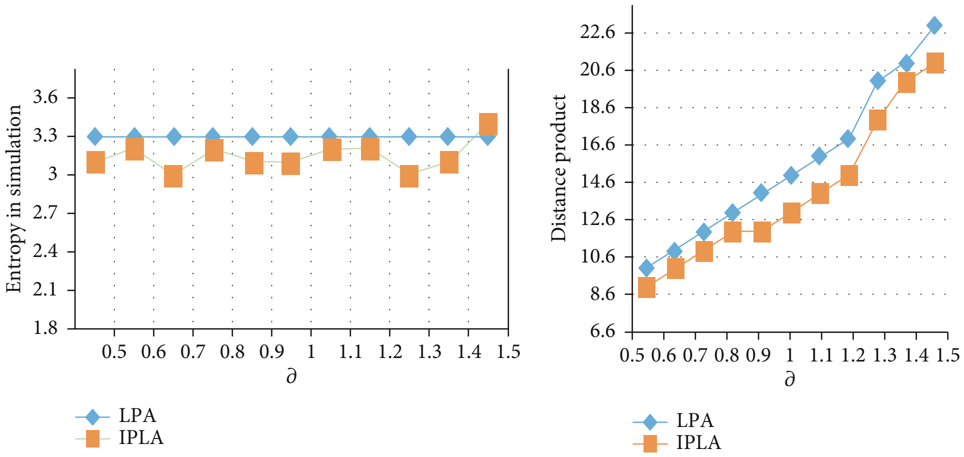


FIGURE 13: Result of experiment about privacy level of various users.

Normal Versus Abnormal. In the second experiment, it is required from participants of group 1 to stimulate a condition which we can label as abnormal condition. To create real abnormal situation, like as tumble, is hard to stimulate. Therefore, to generate abnormal phenomena, some disturbance such as by calling to a participant randomly while the task is being performed, are added in the experiment. After applying disturbed method, our algorithm detects abnormal activity at real-time occurrence without waiting for task's completion as shown in Figure 11(b). So our algorithm detects activity at real time instead of central computing in which abnormality is detected after completion of whole process of activity, and it enhanced the real-time performance. Hence, we found that our scheme protects privacy of user's behavior pattern by detecting abnormal activity at real time without waiting for completion of the process. Table 8 shows the result of abnormal activity detection by our proposed *BehaviorPrivacyDeg*. The



(a) ∂ versus entropy

(b) ∂ versus distance product

Figure 14: Effect of ∂ entropy and product of distance.

transition of participants shown in Figure 9 is $d_{26} \longrightarrow d_{24} \longrightarrow d_{22} \longrightarrow d_{34} \longrightarrow d_{40} \longrightarrow d_{21} \longrightarrow d_{19} \longrightarrow d_{17} \longrightarrow d_{13} \longrightarrow d_{10} \longrightarrow d_{16} \longrightarrow d_{21} \longrightarrow d_{14}$. When abnormality is detected at sensor device d_{19} by interfering the participants, the trajectory remains the same but time duration is significantly changed. Results in Figures 12(a) and 12(b) show the suitability and effectiveness of our scheme.

User Privacy-Level Concern Index. In this section, as discussed in privacy-level design model section, experiment result of our proposed MPCS showed that user's behavior pattern privacy is changed with the changing of privacy level. Privacy levels are configured by using index value on server. In Figure 13, index value showed that most users have much concerned about their activity privacy in smart IoT environment. After this, result revealed that users are more concerned about that area/zone and only 10 percent users are worried about their location. Hence, users can control their privacy level according to their concern by using our proposed MPCS.

Location Privacy. To protect the location of user in smart IoT environment, our proposed LPA achieved privacy of user's location by considering entropy and cloak region (CR). Users are required to share some level of personal information for getting services from server via installed sensor devices which are also called access point (AP).

We used a parameter ∂ to obtain partial information. In our experiments, we used 120 sensor devices which sense data and $\partial = 1.5$ represents the user familiarity about query probability over 75 APs. The effect of ∂ on entropy and product of distance are represented in results of our proposed LPA which is shown in Figures 14(a) and 14(b). In our simulation, $P = 15$, $r = 500$m, and change ∂ is from 0.5 to 1.5. The result revealed that location privacy algorithm (LPA) is better and has achieved the set target. The assessments of results showed that performance of LPA is better.

5. Conclusion

In this paper, we have proposed an effective multilevel privacy controlling scheme based on behavior pattern privacy degree and multilevel privacy design model. To protect the privacy of user's behavior pattern, we introduced *Behavior-PrivacyDeg* based on FPMTA, PCA, and AADTA. *Behavior-PrivacyDeg* focuses to mine, compress, store, and compute activities of user's behavior pattern by using proposed mining, compression algorithms, and storage mechanism. To detect abnormality and to protect the activity, we use the AADTA. Privacy levels are used for controlling method to protect users' behavior pattern. LCPA is used to configure the privacy level of users according to their concern and priority. PLA protects the privacy of user's location. PLA used entropy and cloak region (CR) to ensure privacy of location by spreading fake locations as far as possible. The experiments revealed the performance and feasibility of proposed MPCS. The scheme we proposed could provide a basis for behavior pattern privacy, LBS research, having the practical and theoretical significance on the study of trajectory anonymity, and location-based privacy preserving in smart IoT environment.

Data Availability

There is no data associated with the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of the article.

Authors' Contributions

A.K. and M.M.A.K devised the methodology and acquired funding. A.K. and M.A.J. carried out the formal analysis and data curation. A.K. and M.U.F. wrote the original draft, reviewed the writing, and edited the manuscript. A.A. and C.W. proofread the manuscript before its final submission. A.K, M.M.A.K, and M.A.J. contributed equally to this work.

Acknowledgments

The authors would like to thank the anonymous reviewers for a careful reading of this article and for all their comments, which led to a number of improvements in the article. This work is supported by the National Natural Science Foundation of China under grant No. 61672115, Guangzhou Government Project under grant No. 62104301, and Chongqing Social Undertakings and Livelihood Security Science and Technology Innovation Project Special Program (No. cstc2015jcyjBX0124 and No. cstc2017shmsA30003).

References

[1] P. Kumar, A. Braeken, A. Gurtov, J. Iinatti, and P. H. Ha, “Anonymous secure framework in connected smart home environments,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 968–979, 2017.

[2] K. Shin, X. Ju, Z. Chen, and X. Hu, “Privacy protection for users of location-based services,” *IEEE Wireless Communications*, vol. 19, no. 1, pp. 30–39, 2012.

[3] N. Kaaniche and M. Laurent, “Data security and privacy preservation in cloud storage environments based on cryptographic mechanisms,” vol. 111, Computer Communications Elsevier, 2017.

[4] L. Calderoni, P. Palmieri, and D. Maio, “Location privacy without mutual trust: the spatial Bloom filter,” in *Computer communications*, vol. 68, pp. 4–16, Elsevier, 2015.

[5] R. Roshan and A. K. Ray, “Challenges and risk to implement IOT in smart homes: an Indian perspective,” *International Journal of Computer Applications*, vol. 153, no. 3, pp. 16–19, 2016.

[6] P. Kumar, A. Gurtov, J. Iinatti, M. Ylianttilla, and M. Sain, “Lightweight and secure session-key establishment scheme in smart home environments,” *IEEE Sensors Journal*, vol. 16, no. 1, pp. 254–264, 2016.

[7] B. Chikhaoui, S. Wang, and H. Pigot, “A frequent pattern mining approach for ADLS recognition in smart environments,” in *Proceedings of the 25th IEEE International Conference on Advanced Information Networking and Applications (AINA ’11)*, pp. 248–255, Biopolis, Singapore, March 2011.

[8] S. T. Peddinti, A. Dsouza, and N. Saxena, “Cover locations: availing location-based services without revealing the location,” in *Proceedings of the 10th Annual ACM Workshop on Privacy in the Electronic Society - WPES '11*, New York, NY, USA, 2011.

[9] M. F. Mokbel, C.-Y. Chow, and W. G. Aref, “The new casper: query processing for location services without compromising privacy,” in *Proceedings of the 32nd International Conference on Very Large Data Bases*, Seoul, Korea, 2006.

[10] B. Niu, X. Zhu, W. Li, and H. Li, “Epcloak: an efficient and privacy preserving spatial cloaking scheme for lbss,” in *2014 IEEE 11th International Conference on Mobile Ad Hoc and Sensor Systems*, Philadelphia, PA, USA, 2014.

[11] G. Dini and P. Perazzo, “Uniform obfuscation for location privacy,” in *Data and Applications Security and Privacy XXVI*, Springer, 2012.

[12] C. Y. Chow, M. F. Mokbel, and W. G. Aref, “Casper∗: query processing for location services without compromising privacy,” *ACM Transactions on Database Systems*, vol. 34, no. 4, pp. 1–48, 2009.

[13] K. Fawaz and K. G. Shin, “Location privacy protection for smartphone users,” in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, USA, 2014.

[14] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, “Enhancing privacy through caching in location-based services,” in *2015 IEEE Conference on Computer Communications (INFOCOM)*, Hong Kong, China, 2015.

[15] H. Lu, C. S. Jensen, and M. L. Yiu, “Pad: privacy-area aware, dummy based location privacy in mobile services,” in *Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access - MobiDE '08*, New York, NY, USA, 2008.

[16] L. Fenghua, W. Sheng, N. Ben, and L. H. H. Yuanyuan, “Time obfuscation-based privacy-preserving scheme for location-based services,” *Workshop on Physical-Layer Security: Rise, Fall and Rise Again Trilogy Toward Securing Data Networks*, 2016.

[17] X. Zhang, J. Wang, Q. Gao, X. Ma, and H. Wang, “Device-free wireless localization and activity recognition with deep learning,” in *2016 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pp. 1–5, Sydney, Australia, March 2016.

[18] Z. Liu, “Human activity recognition with 2D and 3D cameras,” in *Progress in Pattern Recognition, Image Analysis, Computer-Vision, and Applications*, L. Alvarez, M. Mejail, L. Gomez, and J. Jacobo, Eds., vol. 7441 of Lecture Notes in Computer Science, p. 37, Springer, Berlin, Germany, 2012.

[19] V. Srinivasan, J. Stankovic, and K. Whitehouse, “Protecting your daily in-home activity information from a wireless snooping attack,” in *Proceedings of the 10th International Conference on Ubiquitous Computing - UbiComp '08*, pp. 202–211, Seoul, South Korea, 2008.

[20] M. Gruteser and D. Grunwald, “Anonymous usage of location based services through spatial and temporal cloaking,” in *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services*, pp. 31–42, New York, NY, USA, 2014.

[21] H. Ngo and J. Kim, “Location privacy via differential private perturbation of cloaking area,” in *28th Computer Security Foundations Symposium*, Verona, Italy, 2015.

[22] G. Natesan and J. Liu, “An adaptive learning model for k- anonymity location privacy protection,” in *39th Annual International Computers, Software and Applications Conference*, Taichung, Taiwan, 2015.

[23] T. Serge, B. Mickala, and Y. Younghwan, “User activity recognition in smart homes using pattern clustering applied to temporal ANN algorithm,” *Sensors*, vol. 15, no. 5, pp. 11953–11971, 2015.

[24] J. Lu and W. Qi Yan, “Comparative Evaluations of Human Behavior Recognition Using Deep Learning,” in *Multimedia Cyber Security Book*, p. 14, IGI Global Publisher of Timely Knowledge, 2020.

[25] M. A. K. Quaid and A. Jalal, “Wearable Sensors Based Human Behavioral Pattern Recognition Using Statistical Features and Reweighted Genetic Algorithm,” in *Multimedia Tools and Applications*, Springer, 2020.

[26] W. Chengliang, Z. Qian, P. Yayun, D. Debraj, and S. Wen-Zhan, “Distributed abnormal activities detection in smart environments,” *International Journal of Distributed Sensor Networks*, vol. 10, Article ID 283197, 2014.

[27] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, San Francisco, CA, USA, 3rd edition, 2011.

[28] M. Rose, M. Delgado, A. Vila, H. Hagras, and A. Bilgin, "A fuzzy logic approach for learning daily human activities in an ambient intelligent environment," in *Proceedings of the IEEE International Conference on bFuzzy Systems (FUZZ-IEEE '12)*, pp. 1–8, Brisbane, QLD, Australia, June 2012.

[29] A. Provotar and A. Lapko, "Fuzzy inference systems and their applications," *Cybernetics and Systems Analysis*, vol. 49, no. 4, pp. 517–525, 2013.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 9984666, 12 pages
https://doi.org/10.1155/2021/9984666

Research Article

BMC-SDN: Blockchain-Based Multicontroller Architecture for Secure Software-Defined Networks

Abdelouahid Derhab,[1] **Mohamed Guerroumi**,[2] **Mohamed Belaoued**,[3] **and Omar Cheikhrouhou**[4]

[1] *Center of Excellence in Information Assurance (CoEIA), King Saud University, Saudi Arabia*
[2] *Faculty of Electronic and Computer Science, USTHB University, Algiers, Algeria*
[3] *LICUS Lab., Department of Computer Science, University of 20 August 1955, Skikda, Algeria*
[4] *College of Computers and Information Technology, Taif University, P.O. Box 11099, Taif 21944, Saudi Arabia*

Correspondence should be addressed to Abdelouahid Derhab; abderhab@ksu.edu.sa

Received 8 March 2021; Revised 21 March 2021; Accepted 31 March 2021; Published 22 April 2021

Academic Editor: Muhammad Shafiq

Multicontroller software-defined networks have been widely adopted to enable management of large-scale networks. However, they are vulnerable to several attacks including false data injection, which creates topology inconsistency among controllers. To deal with this issue, we propose BMC-SDN, a security architecture that integrates blockchain and multicontroller SDN and divides the network into several domains. Each SDN domain is managed by one master controller that communicates through blockchain with the masters of the other domains. The master controller creates blocks of network flow updates, and its redundant controllers validate the new block based on a proposed reputation mechanism. The reputation mechanism rates the controllers, i.e., block creator and voters, after each voting operation using constant and combined adaptive fading reputation strategies. The evaluation results demonstrate a fast and optimal detection of fraudulent flow rule injection.

1. Introduction

Software-defined networks (SDNs) [1] have been widely deployed in many application fields, as they replace the conventional TCP/IP architecture with another one that decouples the networking devices from their control management. This is achieved by moving the network functions that are performed by the network devices to a central entity, which allows an easy and low-cost network management.

To manage large-scale and multidomain networks, multicontroller SDN is proposed [2], where each controller is responsible for one domain. In multicontroller SDN, there are two types of communication: vertical and horizontal. In vertical communication, OpenFlow protocol [3] manages the southbound interface between the controller and the forwarding devices, e.g., switches, by telling switches where to send data flows. The northbound interface manages the communication between the controller and the applications. In horizontal communication, controllers exchange information about network topology between them via their east-west interfaces.

The horizontal communication allows controllers to share any update about network topology like the link state, network devices, changes in the flow table, list of network hosts, and the association between each switch and its controller. Thus, it is important for controllers to maintain the same global network view. To this end, an intercontroller traffic must be exchanged between the controllers, through their east-west interfaces. So, a logical centralized view of the network state must be guaranteed for SDN controllers to facilitate the development of advanced network applications.

The main concern for the network administrator and network programmer is to keep the SDN controllers synchronized and having similar network topology information to make the correct routing decisions. However, multicontroller SDN could be targeted by several attacks [4–6], including false data injection, where a compromised controller sends fraudulent flow information to other controllers. This could cause routing malfunctions, routing loops, and incorrect functionality of firewalls.

To deal with this issue, we propose a security architecture that integrates blockchain technology with multicontroller SDN. The main idea of the architecture is to associate a set of controllers to each domain. Differently from [7] that deploys many controllers for fault-tolerance purposes, our architecture is aimed at ensuring a secure and trustworthy intercontroller communication. To this end, the proposed architecture considers a master controller and redundant controllers for each network domain. Each controller can be master in one domain and redundant in other domains. The master controller creates blocks of network flow updates, and the redundant controllers decide whether to validate the created blocks or not. The architecture also integrates a reputation mechanism that rates the controllers after each voting operations using constant and adaptive fading reputation strategies. In this way, malicious master controllers and redundant controllers that provide incorrect voting will be detected. More specifically, the main contributions of the paper are as follows:

(i) We propose BMC-SDN, a security architecture that integrates SDN and blockchain technologies to secure intercontroller communication. BMC-SDN assigns a single master controller and multiple redundant controllers to each domain. The controllers are members of the blockchain; the master controller creates blocks, and its behavior is monitored by the redundant ones.

(ii) We integrate a reputation mechanism to BMC-SDN, which rates controllers according to two strategies: (1) *constant fading reputation* that allows forgetting past operations of the controller at a constant rate and (2) *combined adaptive fading reputation* that rates the controller using different constants according to the controller's reputation. More precisely, the more the controller misbehaves, the faster positive histories are forgotten. On the other hand, the more the controller well-behaves, the faster negative histories are forgotten.

(iii) We implement the proposed BMC-SDN architecture using different tools such as ONOS, MultiChain, and Mininet software platforms. The evaluation results show that BMC-SDN can detect all injected flows with a low detection delay. In addition, the reputation mechanism allows adaptive detection time of malicious controllers according to the requirements of the network administrator.

The rest of this paper is organized as follows: Section 2 presents related work. The system and threat model is given in Section 3. Section 4 provides a detailed description of BMC-SDN. The implementation and performance evaluation of BMC-SDN are presented in Sections 5 and 6, respectively. Finally, Section 7 concludes the paper.

2. Related Work

Security of SDN has attracted much attention from researchers [6, 8]. Tayfour and Marsono [9] proposed a collaborative technique to detect and mitigate distributed denial-of-service (DDoS) flooding attacks on software-defined network (SDN) across multicontroller domains, using sflow-RT [10] and Snort rules. Halder et al. [11] proposed a mechanism that detects conflicts in distributed SDN controller environment. Each controller generates a directed graph from the forwarding rules. The different graphs are merged to generate a global network state, which allows detecting any kind of flow rule violation and forwarding loops. Das et al. [7] proposed an SDN architecture that replaces the single controller with multiple independent controllers, which allows tolerance to controller failure. Varadharajan et al. [12] proposed a policy-based security architecture for a multidomain SDN network. The packet flows are analyzed to identify an eventual unauthorized flow and dynamically update security policies.

Blockchain has been adopted as an option to secure one-controller SDN. Derhab et al. [13] proposed BICS (Blockchain-based Integrity Checking System), which sends the traffic flow rules of the vSwitch to the blockchain. The fraudulent flow rules are detected if the rules sent by the SDN controller are different from the ones in the blockchain. In [14], the blockchain is also used to secure the communication between the SDN controller and the other network elements against false flow rule injection. In [15], a blockchain mechanism is proposed to protect against unauthorized access and DDoS attack. In this mechanism, the switches are registered and verified using zero proof of knowledge. They are also validated in the blockchain using a voting-based consensus mechanism. Also, a Boltzmann deep learning machine is applied to identify anomalous flow traffic. DistBlockNet [16] verifies and validates the version of the flow rule table using a blockchain and downloads the latest flow rules for the IoT forwarding devices. TrustBlock [17] computes the trust values of SDN nodes based on blockchain. The consensus mechanism of the blockchain is used to filter out dishonest nodes that provide unfair recommendations.

Blockchain is also used to secure multicontroller SDN. Fernando and Wei [18] proposed an infrastructure comprising two layers: (1) multicontroller SDN networking layer and (2) blockchain-based layer. The control/management commands of the SDN controller are hashed and recorded in a smart contract of the blockchain and sent to the targeted SDN controller. The latter verifies the integrity of the command by checking the smart contract. Yang et al. [19] proposed BlockTC, a distributed blockchain-based trusted multidomain collaboration for mobile edge computing, wherein all SDN controllers use the blockchain to obtain topology information of other domains and verify the legitimacy of the routing. Azab et al. [20] proposed a system, where multiple controllers manage the same set of switches.

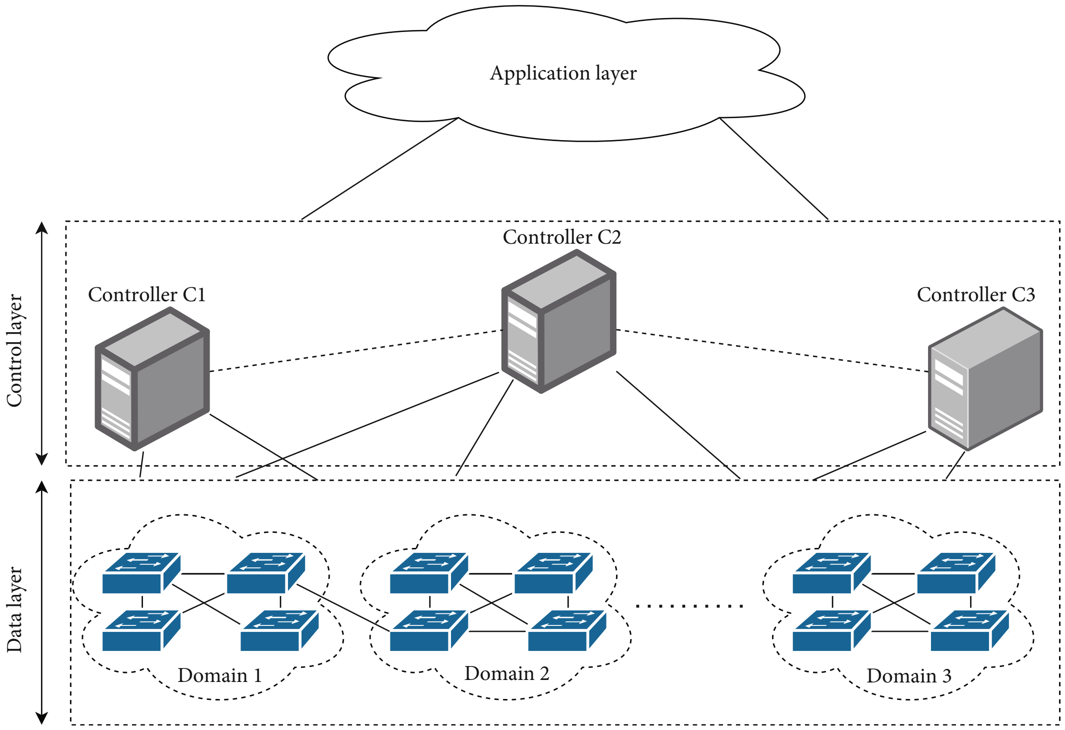


Figure 1: General architecture of multicontroller SDN.

The blockchain is used to ensure consistency between SDN controllers. The switches and the controllers are considered nodes in the blockchain. When the switch broadcasts its request to all controllers, each node in the blockchain can verify and validate the request. Kataoka et al. [21] proposed a trust list that is distributed among IoT devices using blockchain and SDN. The IoT devices can only trust other devices and services that are whitelisted in the trust list.

Blockchain is also proved to be a promising technology to mitigate false data injection in IoT networks. For example, Cheikhrouhou and Koubaa in [22] leveraged blockchain technology to secure localization in IoT networks and guarantee the correctness of the shared information. The proposed solution permits detection of false data injection and therefore reduces the localization error.

3. System and Threat Model

As shown in Figure 1, we consider an SDN network with multiple and distributed controllers. The general SDN architecture contains three layers: application, control, and data layers. The application layer contains programs that explicitly communicate their network requirements and desired network rules to the controllers. The control layer contains N controllers deployed in a distributed manner. The data layer contains devices that are configured in N different domains. Each domain is controlled by one master controller, and each master controller has more than one slave or redundant controllers. In addition to its primary role, the master controller is configured as a redundant controller for more than one domain. The controllers in a distributed multicontroller SDN keep the same network global view (i.e., same topology and same link status). Any change in the state of each controller such as new flow configurations and link failures must be quickly sent to other controllers. If the controllers have an inconsistent view, the network strategies may not be executed correctly, which could cause network misconfigurations such as routing loops, packet losses, and firewall leaks.

Figure 2 shows the possible threats and attacks that could occur in a multicontroller SDN architecture. If the communication between controller C1 and the other controllers fails, then the network could be partitioned. In this case, if the network topology changes in the domain of controller C1, the other controllers cannot be informed and vice versa. Therefore, the controllers could make routing decisions based on their own old view of the network topology, which could lead to unforeseen events.

Moreover, an attacker can intercept the communication between controllers and injects false data. This man-in-the-middle attack (Figure 2: arrow 1) can lead to an inconsistent network view, which leads to routing errors such as routing loops (Figure 2: arrow 2), firewall leaks, poor and false routing decision, and congestion of critical links when the majority of the false flow rules go through the same link, which ultimately affects the performance of the network.

In addition, it is possible for an attacker in this communication model to spoof controllers (Figure 2: arrow 3) and send false information about the topology, the state of the links, and the hosts of each domain. This false information can cause several problems such as traffic congestion, device overloading, and wrong routing information.

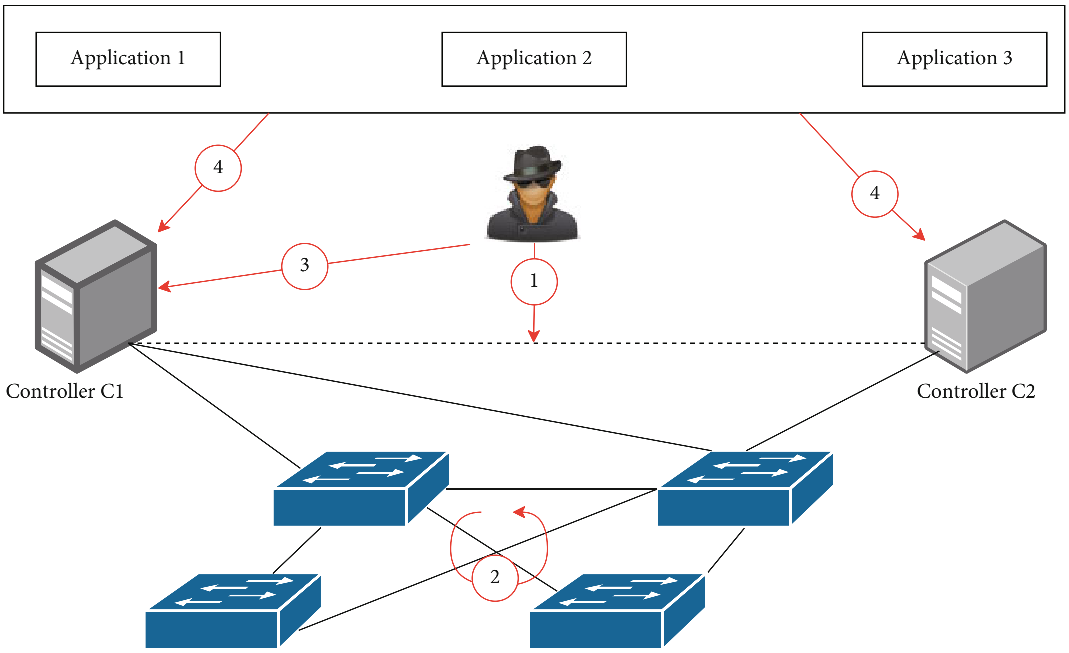


Figure 2: Threat model for multicontroller SDN architecture.

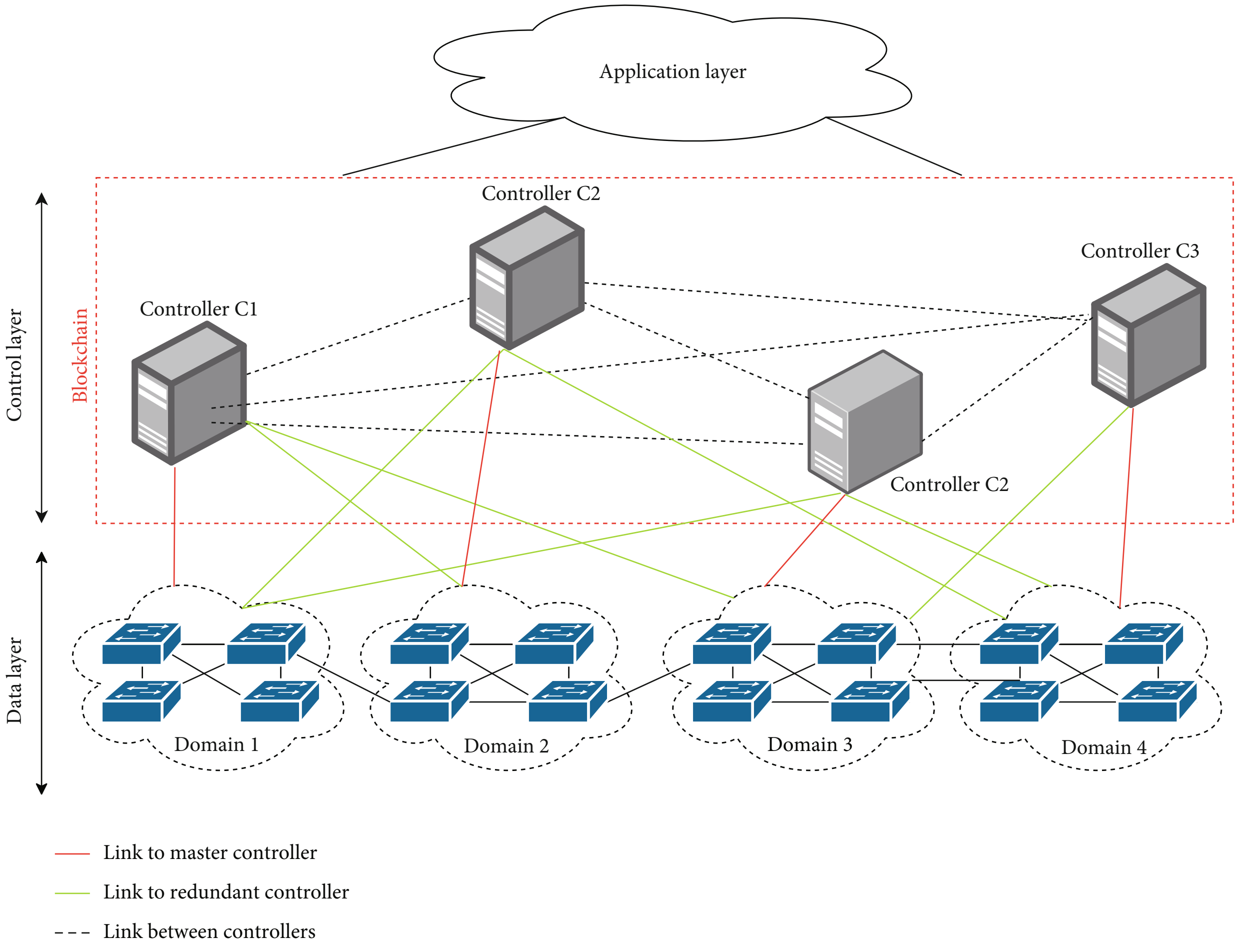


Figure 3: The proposed BMC-SDN architecture.

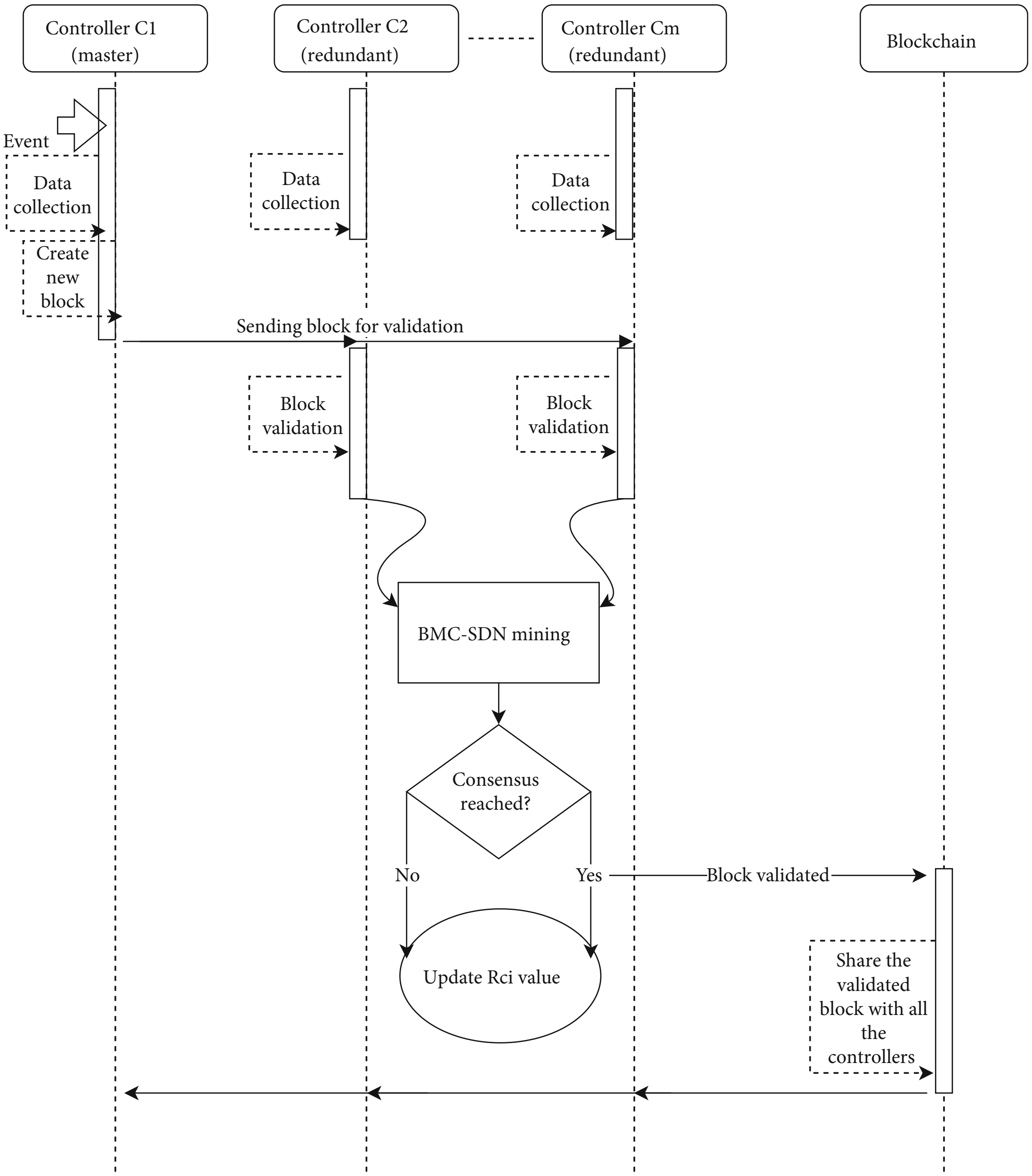


Figure 4: BMC-SDN sequence diagram.

Another critical attack could be launched by exploiting some APIs that are offered by the SDN controller [23]. These APIs allow developers to implement new network control applications. Indeed, if an application is compromised (Figure 2: arrow 4), the security of the entire network can be compromised. Such an attack can have dramatic consequences such as the modification of the behavior of the network and the intercepting of the network traffic, which can trigger large distributed denial-of-service attacks [24].

4. Blockchain-Based Multicontroller Software-Defined Network (BMC-SDN)

In this section, we provide a description of the proposed BMC-SDN architecture. The aim of BMC-SDN is to protect the control layer of the SDN architecture, previously presented in Figure 1, against the different attacks previously discussed in Section 3. To this end, BMC-SDN uses blockchain to protect the communication among controllers. Figure 3 gives a general overview of our proposed BMC-SDN solution. The control layer is protected thanks to blockchain. More precisely, all controllers are members of a blockchain network, and they communicate with each other via this blockchain. In BMC-SDN, we focus on the security of the control layer and the east-west communication of this layer. For the security of the communication between the controllers and the data layer devices, we consider our work in [13, 14]. Let N be the number of controllers in the network. For each domain, we select a single master controller and M redundant controllers, $2 \leq M < N$. The redundant controllers replace the master controller in case of failure. A redundant controller cannot replace several master controllers unless it is the only available redundant one. The selection of the redundant controller that will replace the master controller is made according to its identity. More precisely, the redundant controller with the smallest ID is selected. In addition, the M redundant controllers of the same domain monitor the behavior of their master controller and participate in the consensus of validating the data blocks created by the master controller.

Figure 4 shows the sequence diagram that explains the main steps and operations in BMC-SDN. The master controller is responsible for orchestrating the traffic of its domain, and its redundant controllers monitor the domain, receive the same events, and apply the same updates as the master controller but have no influence on the domain. A redundant controller can control the domain in case the master controller fails as already explained. The controller has a local structure that contains OpenFlow commands [3] and local information such as network topology, list of hosts, and link state. This information could be changed due to several events such as topology change, link failure, and device failure. In case of any change, the master controller receives the corresponding event, performs the required update, and informs the other controllers. So, the master controller creates a block containing these updates and sends it through the blockchain to its redundant controllers. The latter, which have already received the same event as the master, validate this block according to the following consensus protocol: If the number of redundant controllers validating the block exceeds a threshold S, then the consensus is reached. In this case, the block is considered valid and will be added to the blockchain. This allows all controllers to have a global view of the entire network. If the consensus is not reached, the block is considered invalid. In this case, the master controller and the redundant controllers who validated this block will be negatively rated, as explained in Section 4.2.

4.1. BMC-SDN Trusted Node. The trusted node in BMC-SDN has the read and write permissions on the blockchain. All the master controllers are considered trusted nodes. They can read from blockchain and create new blocks. The creation of a new block is triggered by receiving a new external event that is sent from the data layer. When a master controller receives new information from the data layer devices of its domain, like flow rule request, it creates a new block containing adequate information and shares this new block with the redundant controllers for validation. The validated block is shared with all controllers in the network. Therefore, each controller can build the same global network view.

4.2. BMC-SDN Consensus and Reputation Mechanism. The consensus process is performed by the redundant controllers. These controllers are considered miners. They are responsible for validating newly created blocks. After creating a new block by the master controller, the new invalid block is shared with the miners. The miners start the validation process by comparing the similarity between the information contained in the invalid block and their local information. The miners receive the same request as the master controller and then generate the needed reply. For example, they generate the same flow rule for the same flow rule request. So, the miner can check the similarity and accordingly validate the new block. Upon validation, the new block will be added to the chain. Miners who have a different opinion than the majority and the master controller whose block has not been validated could be considered a malicious controller. The detection of the malicious controller will be calculated based on the below reputation mechanism.

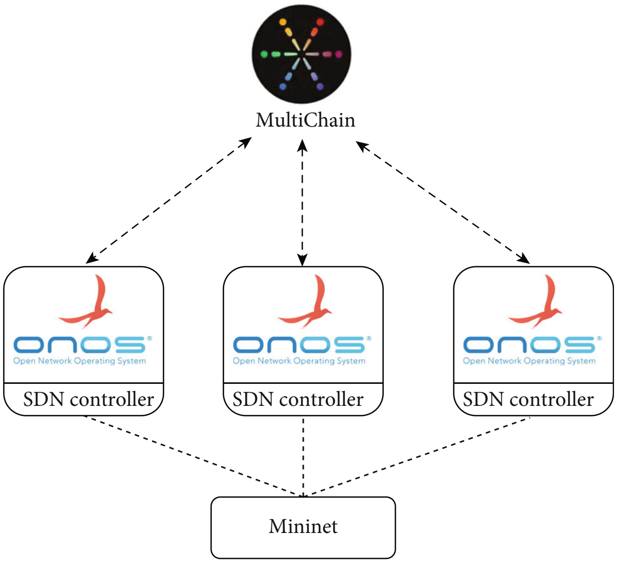

MultiChain
onos
Open Network Operating System
SDN controller
onos
Open Network Operating System
SDN controller
onos
Open Network Operating System
SDN controller
Mininet

Figure 5: Implementation of BMC-SDN architecture.

Table 1: SDN implementation parameters.

Parameter	Value
Number of SDN domains	3
Number of redundant controllers	2
Number of switches	[10-100]
Number of hosts	[10-450]

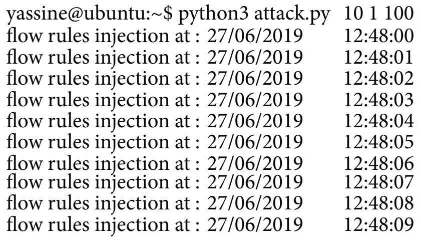
This reputation mechanism is seen as an additional layer of security to protect the control layer and therefore the entire network. This mechanism is based on the reputation management of controllers. Each controller C_i has a reputation value R_i shared between miners in the same blockchain. The value of R_i is between 0 and 1 ($0 \leq R_i \leq 1$). In this mechanism, each controller can be in one of the following three states, depending on its reputation value R_i:

(i) Trustworthy controller, if $R_i \geq 0.8$. In this case, the information sent by the controller is evaluated and taken into account by the miners.

(ii) Suspicious controller, if $0.5 \leq R_i < 0.8$. In this case, the information sent by the controller is evaluated but are not taken into account by the miners.

Table 2: Description of the data structures.

Data stores	Description
Mastership store	Associates between each switch and its master.
Network topology store	Describes the network topology in terms of links and switches.
Flow store	Saves the flows of each switch from the master controller to the slave controller, when a change in the flow table is detected.
Host store	Manages the list of network hosts.
Application store	Manages the application inventory.
Intent store	Manages the inventory of intentions. Intentions are part of the ONOS intention framework used by applications to define the network policy, without specifying in detail how the data plan should actually be programmed.
Component configuration store	Stores system-wide configurations for various software components in ONOS.
Network configuration store	Stores the inserted network configurations in ONOS.
Security mode store	Manages the authorizations granted to applications using the RAFT protocol [33]. Instead, security violations are handled using antientropy protocol.



```
yassine@ubuntu:~$ python3 attack.py  10 1 100
flow rules injection at : 27/06/2019     12:48:00
flow rules injection at : 27/06/2019     12:48:01
flow rules injection at : 27/06/2019     12:48:02
flow rules injection at : 27/06/2019     12:48:03
flow rules injection at : 27/06/2019     12:48:04
flow rules injection at : 27/06/2019     12:48:05
flow rules injection at : 27/06/2019     12:48:06
flow rules injection at : 27/06/2019     12:48:07
flow rules injection at : 27/06/2019     12:48:08
flow rules injection at : 27/06/2019     12:48:09
```

Figure 6: Attack experiment.

(iii) Malicious controller, if $R_i < 0.5$. The communication traffic of this controller is ignored by others, until intervention of the network administrator. When $R_i < 0.5$, the reputation of C_i is continuously updated and switches to the suspicious and the trustworthy states when $R_i \geq 0.5$ and $R_i \geq 0.8$, respectively.

The miner controllers evaluate the controller C_i according to the consensus result:

(i) If the consensus is reached, the block sent by the master controller will be validated and the value of its reputation increases.

(ii) If the consensus is not reached, the block sent by the master controller will not be validated, and thus, the value of its reputation decreases.

(iii) For miners who have the same opinion as the majority, their reputation will increase.

(iv) For miners who have a different opinion than the majority, their reputation will decrease.

More precisely, the value of R_i is computed as follows:

(i) We compute the reputation of controller C_i, denoted by RP_i, during each time period (or observation interval). Formally, $RP_i = P_i / TP_i$, where P_i is the number of positive participation and TP_i is the total number of positive participation made by controller C_i in blockchain operations (creation and validation of blocks).

(ii) If $RP_i < 0.5$, the reputation of C_i during the current time period RT_i is set to 0. Otherwise, it is set to 1.

(iii) $R_i = \omega R_i + (1 - \omega) RT_i$, where $\omega \in [0, 1]$ is a constant fading (or discount) factor for past participations.

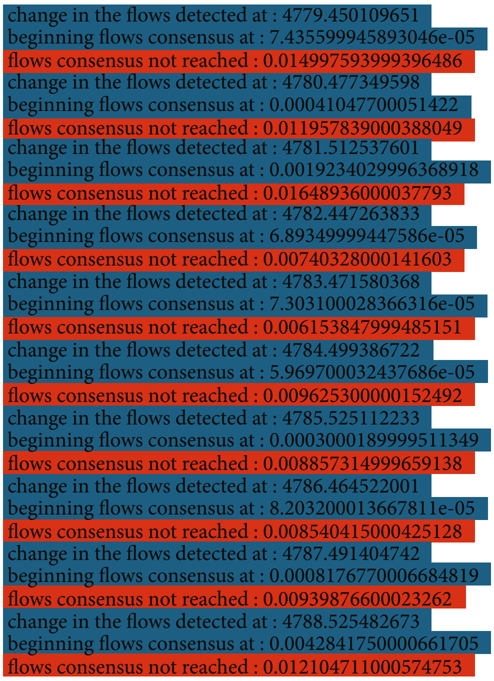
By using the constant fading factor, both positive and negative histories are forgotten at the same rate. Let us consider the following scenario:

(i) If ω is low, we have two scenarios:

(a) If the controller is trustworthy and starts behaving maliciously, then the positive history will be forgotten slowly, and hence, detection time of the controller will be high.

(b) If the controller is malicious and starts well-behaving, then the negative history will be forgotten slowly, and hence, redemption time of the controller will be high.

(ii) If ω is high, we have two scenarios:



```
change in the flows detected at : 4779.450109651
beginning flows consensus at : 7.4355999945893046e-05
flows consensus not reached : 0.014997593999396486
change in the flows detected at : 4780.477349598
beginning flows consensus at : 0.00041047700051422
flows consensus not reached : 0.011957839000388049
change in the flows detected at : 4781.512537601
beginning flows consensus at : 0.0019234029996368918
flows consensus not reached : 0.01648936000037793
change in the flows detected at : 4782.447263833
beginning flows consensus at : 6.89349999447586e-05
flows consensus not reached : 0.00740328000141603
change in the flows detected at : 4783.471580368
beginning flows consensus at : 7.303100028366316e-05
flows consensus not reached : 0.006153847999485151
change in the flows detected at : 4784.499386722
beginning flows consensus at : 5.969700032437686e-05
flows consensus not reached : 0.009625300000152492
change in the flows detected at : 4785.525112233
beginning flows consensus at : 0.0003000189999511349
flows consensus not reached : 0.008857314999659138
change in the flows detected at : 4786.464522001
beginning flows consensus at : 8.203200013667811e-05
flows consensus not reached : 0.008540415000425128
change in the flows detected at : 4787.491404742
beginning flows consensus at : 0.0008176770006684819
flows consensus not reached : 0.00939876600023262
change in the flows detected at : 4788.525482673
beginning flows consensus at : 0.004284175000661705
flows consensus not reached : 0.012104711000574753
```

Figure 7: Detection experiment.

(a) If the controller is trustworthy and starts behaving maliciously, then the positive history will be forgotten quickly, and hence, detection time of the controller will be low.

(b) If the controller is malicious and starts well-behaving, then the negative history will be forgotten quickly, and hence, redemption time of the controller will be low. In this scenario, the controller could take advantage of the reputation system and act maliciously most of the time, and when its state becomes suspicious or malicious, it just needs to perform few positive operations to return back to the trustworthy state.

Based on the above scenarios, we can observe that using a constant fading factor has some disadvantages. To deal with this issue, we propose using different fading factors according to the reputation of the controller, as follows:

$$R_i = \begin{cases} \omega_3 R_i + (1-\omega_3) RT_i, \text{ when } R_i \geq 0.8, \\ \omega_2 R_i + (1-\omega_2) RT_i, \text{ when } 0.5 \leq R_i < 0.8, \\ \omega_1 R_i + (1-\omega_1) RT_i, \text{ when } R_i < 0.5, \end{cases} \quad (1)$$

where $\omega_3, \omega_2, \omega_1 \in [0, 1]$ and $\omega_3 > \omega_2 > \omega_1$. The above equation uses combined fading factors, i.e., the more the controller misbehaves, the faster (resp., the slower) positive histories (resp., negative histories) are forgotten. On the other hand, the more the controller well-behaves, the faster (resp., the slower) negative histories (resp., positive histories) are forgotten.

Table 3: Detection rate vs. number of attacks.

Number of attacks	Detection rate (%)
10	100
20	100
30	100
40	100
50	100
60	100
70	100
80	100
90	100
100	100

5. Implementation

5.1. Implementation Environment. In this section, we implement the blockchain-based secure multicontroller architecture, as shown in Figure 5, using the following components:

(1) *SDN Controller*. We use the Open Network Operating System (ONOS) [25] to implement the SDN controller. It provides the control plane that supports the deployment of several controllers as a domain. The implementation parameters of the SDN controller are shown in Table 1.

(2) *Blockchain*. We use MultiChain [26], an open source platform to implement a private blockchain. It can assign privileges to nodes and control who can connect, send, and receive transactions and who can create flows and blocks. Each MultiChain node is accessible through the MultiChain Web Demo [27], which is a simple web interface for multiChain blockchains.

(3) *Mininet* [28]. It creates a virtual network supporting OpenFlow [3] and consisting of switches and real applications that are deployed on a single machine (virtual or real machine or cloud).

The code that we used to implement BMC-SDN can be found in [29]. In addition, we use other tools to implement our solution, such as Postman [30], an application that allows sending http requests and manage authentication. We also use SecureCRT [31], which is software for network administration and end-user access. Our solution is implemented using Python programming language, and some libraries like HTTPBasicAuth, and Requests that authenticate and interact with REST APIs of the ONOS SDN controllers, respectively. We also use JSON (i.e., JavaScript Object Notation) [32] in order to represent data that are processed by the controller data.

5.2. Data Structures. We handle data stores, which are the real distributed data structures in ONOS controllers; the main ONOS Stores are presented in Table 2.

Among the distributed stores presented in Table 2, there are those that are related to the behavior of the data plane such as the network topology store, the flow store, and the host store. The other distributed stores are considered application-specific.

6. Performance Evaluation

In this section, we evaluate the performance of BMC-SDN using the following metrics:

(1) *Total Execution Time*. It represents the required time to transfer a flow through the blockchain, denoted by (T_{Total}). It is the sum of three elements, relating to the number of switches and the number of hosts in the network, namely, (1) the consensus time, (2) the time to send a block, and (3) the time to update data:

$$T_{\text{Total}} = T_{\text{Consensus}} + T_{\text{Sent}} + T_{\text{Update}}. \quad (2)$$

(2) *Detection Rate (DR)*. It represents the ratio of the number of detected attacks to the total number of injected attacks.

(3) *Detection Time (DT)*. It records the required time to detect malicious controllers.

To assess the robustness of our BMC-SDN solution, we inject fraudulent flows at the controller, as depicted in Figure 6. In Figure 7, we can see that these flows are detected as malicious and reported to the administrator by adding an entry to the log file, containing details of the detected anomaly.

Table 4: Execution times vs. number of switches.

Number of switches	CTC (s)	TTBC (s)	UTBC (s)	TT (s)
10	0.019	0.007	0.018	0.055
20	0.037	0.012	0.013	0.062
30	0.054	0.013	0.02	0.087
40	0.044	0.016	0.035	0.095
50	0.059	0.019	0.025	0.103
60	0.108	0.035	0.049	0.192
70	0.089	0.053	0.059	0.201
80	0.109	0.093	0.079	0.281
90	0.15	0.074	0.099	0.323
100	0.193	0.053	0.087	0.333

CTC: consensus time; TTBC: transfer time within the blockchain; UTBC: update time from the blockchain; TT: total time.

Table 5: Execution times vs. number of hosts.

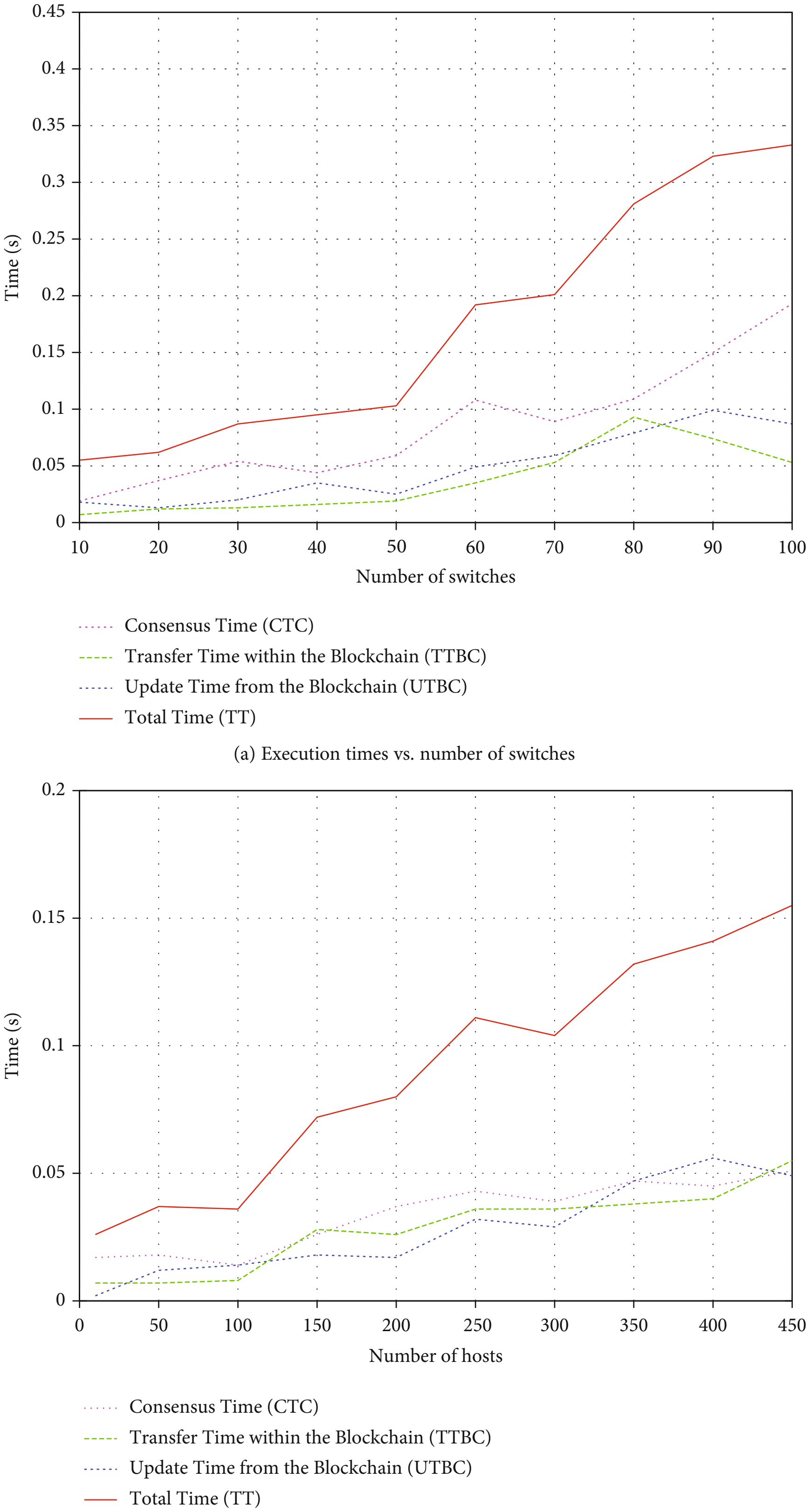
Number of hosts	CTC (s)	TTBC (s)	UTBC (s)	TT (s)
10	0.017	0.007	0.002	0.026
50	0.018	0.007	0.012	0.037
100	0.014	0.008	0.014	0.036
150	0.026	0.028	0.018	0.072
200	0.037	0.026	0.017	0.08
250	0.043	0.036	0.032	0.111
300	0.039	0.036	0.029	0.104
350	0.047	0.038	0.047	0.132
400	0.045	0.04	0.056	0.141
450	0.051	0.055	0.049	0.155

CTC: consensus time; TTBC: transfer time within the blockchain; UTBC: update time from the blockchain; TT: total time.

Table 3 represents the detection rate according to the number of injected attacks. As shown in the table, BMC-SDN ensures the detection rate of 100%, which indicates success in detecting all the injected attacks in the network. As the redundant controllers have the same network view as the master controller, any fraudulent flow that is injected by the master is detected by the redundant ones during the block validation.

As shown in Tables 4 and 5 and Figure 8, we can observe that the total execution time increases as the number of switches increases. We can also observe that the execution time of the consensus increases when the number of switches and hosts is increased. However, the recorded values of execution time are very low.

Figure 9 shows the detection time of the reputation mechanism when the controller behaves maliciously under three values of constant fading factors $\omega = 0.2,0.5,0.8$ and under the combined fading factor such that $\omega_3 = 0.8$, $\omega_2 = 0.5$, and $\omega_1 = 0.2$. We can observe that the reputation of the



(a) Execution times vs. number of switches

(b) Execution times vs. number of hosts

Figure 8: Execution time.

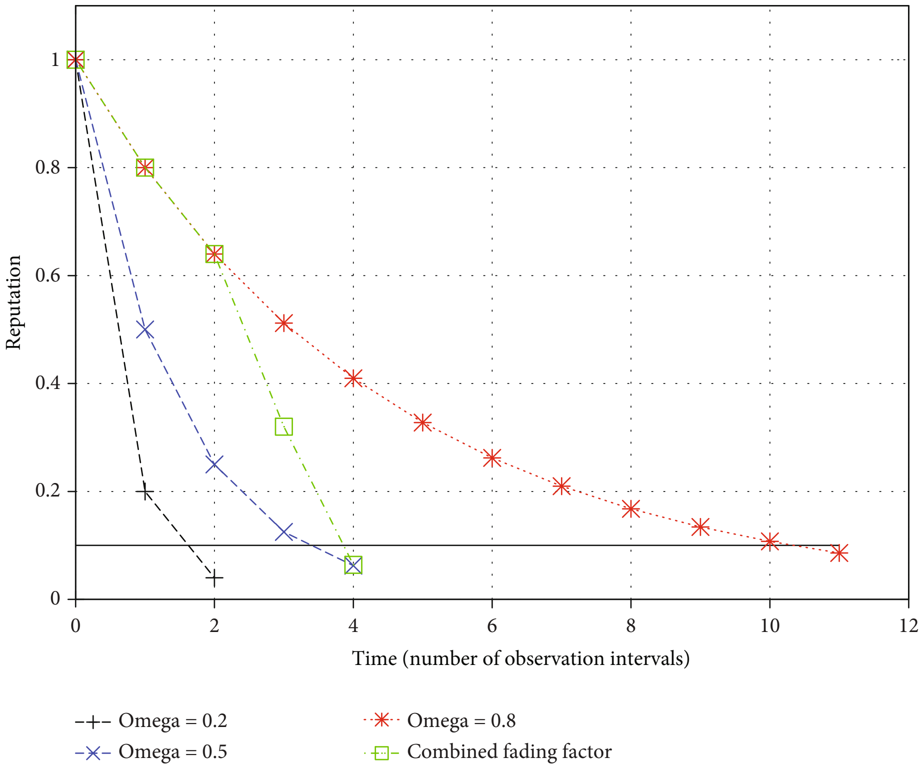


Figure 9: Detection time of reputation mechanism.

controller decreases slowly under high constant fading factor, and hence, high detection time is achieved (i.e., $\omega = 0.8$), and decreases quickly under low constant fading factor, and hence, low detection time is achieved (i.e., $\omega = 0.2$). We can also observe that the combined fading factor employs different fading factors depending on the reputation of the controller. When $R_i \geq 0.8$, it slowly decreases when the fading factor is high (i.e., $\omega = 0.8$). When $R_i < 0.8$, it decreases at a higher speed, and hence, lower detection time is achieved.

7. Conclusion

In this paper, we have proposed BMC-SDN, a blockchain-based multicontroller architecture for secure software-defined networks. In this architecture, we cluster network devices into SDN domains. One master controller and multiple redundant controllers are assigned to each SDN domain. We have used a blockchain where the master controller creates blocks of network flow updates, and redundant controllers validate the blocks. After each voting operation, a reputation mechanism is invoked to rate the controllers, i.e., block creator and voters. The reputation mechanism employs constant and adaptive combined fading reputation strategies to manage and customize the detection time of malicious controllers. The proposed security architecture has been implemented and tested using ONOS, MultiChain, and Mininet software platforms. The evaluation results have reached 100% detection of flow rule injections in a short time. In addition, the proposed combined-fading reputation mechanism has allowed adaptive configuration of fading parameters to reach desired detection time. As BMC-SDN only considers the security of east-west interfaces, we plan as future work to cover the rest of the security planes of SDN architecture, especially the southbound interfaces.

Data Availability

We used data generated from attack scripts and simulators.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

Dr. Omar Cheikhrouhou thanks Taif University for its support under the project Taif University Researchers supporting project number (TURSP-2020/55), Taif University, Taif, Saudi Arabia.

References

[1] F. Hu, Q. Hao, and K. Bao, “A survey on software-defined network and openflow: from concept to implementation,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 2181–2206, 2014.

[2] T. Hu, Z. Guo, P. Yi, T. Baker, and J. Lan, “Multi-controller based software-defined networking: a survey,” *IEEE Access*, vol. 6, pp. 15980–15996, 2018.

[3] 2021, https://www.sdxcentral.com/sdn/definitions/what-is-openflow/.

[4] M. Imran, M. H. Durad, F. A. Khan, and A. Derhab, “Toward an optimal solution against denial of service attacks in software

defined networks," *Future Generation Computer Systems*, vol. 92, pp. 444–453, 2019.

[5] W. LI, W. MENG, Z. LIU, and M. H. AU, "Towards blockchain-based software-defined networking: security challenges and solutions," *IEICE Transactions on Information and Systems*, vol. E103.D, no. 2, pp. 196–203, 2020.

[6] Z. Shah and S. Cosgrove, "Mitigating ARP cache poisoning attack in software-defined networking (SDN): a survey," *Electronics*, vol. 8, no. 10, p. 1095, 2019.

[7] R. K. Das, F. H. Pohrmen, A. K. Maji, and G. Saha, "FT-SDN: a fault-tolerant distributed architecture for software defined network," *Wireless Personal Communications*, vol. 114, no. 2, pp. 1045–1066, 2020.

[8] R. Swami, M. Dave, and V. Ranga, "Software-defined networking-based DDoS defense mechanisms," *ACM Computing Surveys*, vol. 52, no. 2, pp. 1–36, 2019.

[9] O. E. Tayfour and M. N. Marsono, "Collaborative detection and mitigation of distributed denial-of-service attacks on software-defined network," *Mobile Networks and Applications*, vol. 25, no. 4, pp. 1338–1347, 2020.

[10] M. Afaq, S. Rehman, and W.-C. Song, "Large flows detection, marking, and mitigation based on sFlow standard in SDN," *Journal of Korea Multimedia Society*, vol. 18, no. 2, pp. 189–198, 2015.

[11] B. Halder, M. S. Barik, and C. Mazumdar, "Detection of flow violation in distributed SDN controller," in *2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT)*, pp. 1–6, Kolkata, 2018.

[12] V. Varadharajan, K. Karmakar, U. Tupakula, and M. Hitchens, "A policy-based security architecture for software-defined networks," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 897–912, 2019.

[13] A. Derhab, M. Guerroumi, A. Gumaei et al., "Blockchain and random subspace learning-based IDS for SDN-enabled industrial IoT security," *Sensors*, vol. 19, no. 14, p. 3119, 2019.

[14] S. Boukria, M. Guerroumi, and I. Romdhani, "BCFR: blockchain-based controller against false flow rule injection in SDN," in *2019 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1034–1039, Barcelona, Spain, 2019.

[15] M. Singh, G. S. S. Aujla, A. Singh, N. Kumar, and S. Garg, "Deep-learning-based blockchain framework for secure software-defined industrial networks," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 606–616, 2021.

[16] P. K. Sharma, S. Singh, Y.-S. Jeong, and J. H. Park, "DistBlockNet: a distributed blockchains-based secure SDN architecture for IoT networks," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 78–85, 2017.

[17] B. Zhao, Y. Liu, X. Li, J. Li, and J. Zou, "TrustBlock: an adaptive trust evaluation of SDN network nodes based on double-layer blockchain," *PloS one*, vol. 15, no. 3, article e0228844, 2020.

[18] P. Fernando and J. Wei, "Blockchain-powered software defined network-enabled networking infrastructure for cloud management," in *2020 IEEE 17th Annual Consumer Communications & Networking Conference (CCNC)*, pp. 1–6, Las Vegas, NV, USA, 2020.

[19] H. Yang, Y. Liang, J. Yuan, Q. Yao, A. Yu, and J. Zhang, "Distributed blockchain-based trusted multidomain collaboration for mobile edge computing in 5G and beyond," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 11, pp. 7094–7104, 2020.

[20] M. Azab, R. R. Ergawy, E. M. Ghourab, A. Mokhtar, and M. Rizk, "Towards blockchain-based multi-controller managed switching for trustworthy SDN operation," in *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 0991–0998, Vancouver, BC, Canada, 2019.

[21] K. Kataoka, S. Gangwar, and P. Podili, "Trust list: Internet-wide and distributed IoT traffic management using blockchain and SDN," in *2018 IEEE 4th world forum on internet of things (WF-IoT)*, pp. 296–301, Singapore, 2018.

[22] O. Cheikhrouhou and A. Koubaa, "BlockLoc: secure localization in the internet of things using blockchain," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 629–634, Tangier, Morocco, 2019.

[23] O. Cheikhrouhou, "Secure group communication in wireless sensor networks: a survey," *Journal of Network and Computer Applications*, vol. 61, pp. 115–132, 2016.

[24] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: a survey, some research issues, and challenges," *IEEE communications surveys & tutorials*, vol. 18, no. 1, pp. 602–622, 2016.

[25] 2021, https://www.opennetworking.org/onos/.

[26] 2021, https://www.multichain.com/.

[27] 2021, https://github.com/MultiChain/multichain-web-demo.

[28] 2021, http://mininet.org/.

[29] Bmc-sdn code. https://github.com/medguerroumi/BMC-SDN/blob/35a61658700b1fe2d791b8d4f75d2edaa872839f/README.md.

[30] 2021, https://www.postman.com/.

[31] 2021, https://www.vandyke.com/products/securecrt/.

[32] 2021, https://www.jsonrpc.org/specification.

[33] The raft consensus algorithm. https://raft.github.io/.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5529655, 7 pages
https://doi.org/10.1155/2021/5529655

Research Article

Modulation of the Transmission Spectra of the Double-Ring Structure by Surface Plasmonic Polaritons

Senfeng Lai,[1] Yanpei Guo,[1] Guiyang Liu,[1] Chun Shan,[1] Lixin Huang,[1] Yicong Zhang,[1] Yanghui Wu,[2] Wenhua Gu,[2] and Wen Wu[2]

[1] *School of Electronic and Information, Guangdong Polytechnical Normal University, Guangzhou 510635, China*
[2] *School of Electronic and Optical Engineering, Nanjing University of Science and Technology, Nanjing 210094, China*

Correspondence should be addressed to Chun Shan; shanchun@gpnu.edu.cn

Received 5 February 2021; Revised 22 February 2021; Accepted 3 April 2021; Published 15 April 2021

Academic Editor: zhihong tian

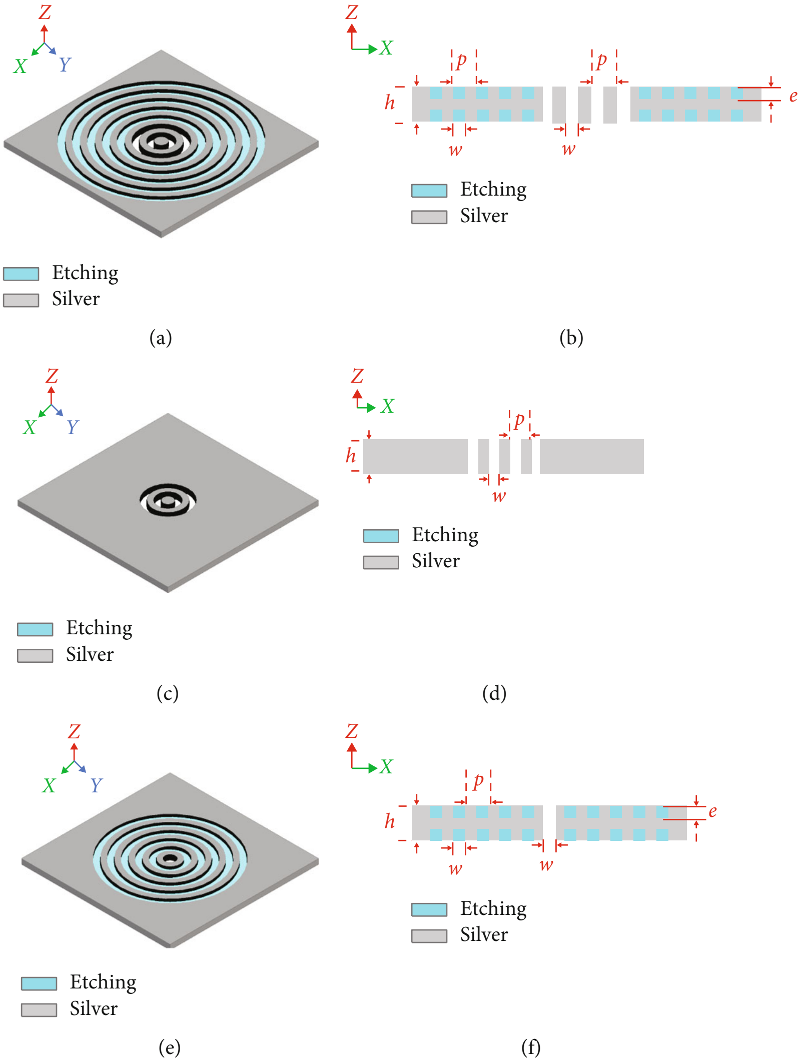
This paper proposes a new structural design to excite surface plasmonic polaritons to enhance the double-ring interference structure. The double-ring structure was etched into a thin film to form fundamental interference patterns, and periodic concentric-ring grooves were employed to gather energy from the surrounding regions through the excitation of surface plasmonic polaritons. Accordingly, the energy of the incident light can be concentrated at the center. The surface plasmon modulates the interference pattern and the transmission spectra. The transmission peak position and its intensity can be tuned by changing the alignment of the grooves. The proposed structure can be applied for designing plasmonic devices as useful components of the plasmonic toolbox.

1. Introduction

With the development of Internet of things and cloud technology, the amount of data that modern communication needs to process is becoming more and more huge [1–5]. In order to improve the communication bandwidth, more and more communication networks choose optical fiber transmission. In the optical transmission, how to increase the light intensity has become a hot topic. The double-slit interference experiment has been known as one of the most profound experiments in physics since the last century. However, surface plasmonic polaritons (SPPs), as a hot research topic owing to their energy enhancement effect at a specific wavelength, are a relatively new concept in physics. The combination of surface plasmonic polaritons and double-slit interference is an exciting research topic [6–20]. The outcomes have potential applications in quantum physics, fundamental optics, optical imaging, detection, integrated circuit design, and other fields [21–28].

In this paper, a new double-ring structure, as shown in Figures 1(a) and 1(b), is proposed to combine the double-slit interference with SPPs to explore exciting possibilities. The proposed structure is a double-ring structure formed by the combination of the periodic concentric-ring grooves [7, 15] and a modified version of Young's double-slit structure and allows for a strong interaction between the two substructures. In the structures shown in Figures 1(a) and 1(b), two concentric rings are shown etched into a silver thin film to make scope for a fundamental double-ring interference. A series of periodic concentric-ring grooves was etched into both the upper and lower surfaces of a silver film to stimulate surface plasmonic polaritons.

Simulations were used to study the transmitted interference pattern and the transmission spectrum of the proposed structure based on the finite difference time domain (FDTD) method. The standout feature of the proposed structure is that it makes use of the classic bull's eye structure to efficiently gather the energy from the surrounding areas and supply it to the center of the interference pattern. It also contains many geometric factors for adjusting and fine-tuning the transmission results. Both the interference pattern and the transmission spectrum were studied.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 1: Schematic diagram of the proposed structure: double-ring aperture at the center of the silver film surrounded by periodic concentric-ring grooves, top view (a) and side view (b); schematic diagram of the reference aperture-only structure: double-ring aperture at the center of the silver film, top view (c) and side view (d); schematic diagram of the bull's eye structure, top view (e) and side view (f).

2. Precise Modulation of the Transmission Spectrum of the Concentric-Ring Interference

The proposed structure is displayed in Figures 1(a) (top view) and 1(b) (side view). The silver thin film consists of two concentric-ring apertures at the center, and a series of periodic concentric-ring grooves was etched into the thin film from both sides. The concentric-ring pitch was $p = 600$ nm and the width was $w = 300$ nm, for the center apertures as well as the surrounding grooves. The groove depth was $e = 60$ nm on each side, and the thickness of the thin film before etching was $h = 300$ nm. The reference aperture-only structure was also studied as a comparison, as shown in Figures 1(c) (top view) and 1(d) (side view). The standard bull's eye structure, which contained only one round aperture at the center as shown in Figures 1(e) (top view) and 1(f) (side view), had the same concentric-ring grooves in the surroundings as in Figures 1(a) and 1(c). The radius of the central aperture was 300 nm. The surrounding medium was air. All the other conditions were kept identical for the above three structures. The incident light was a plane wave transmitted under the thin film from the $-z$-direction to the $+z$-direction. The finite difference time-domain (FDTD) method

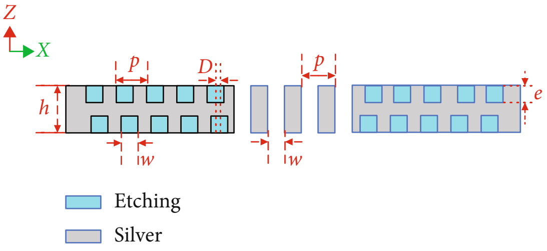
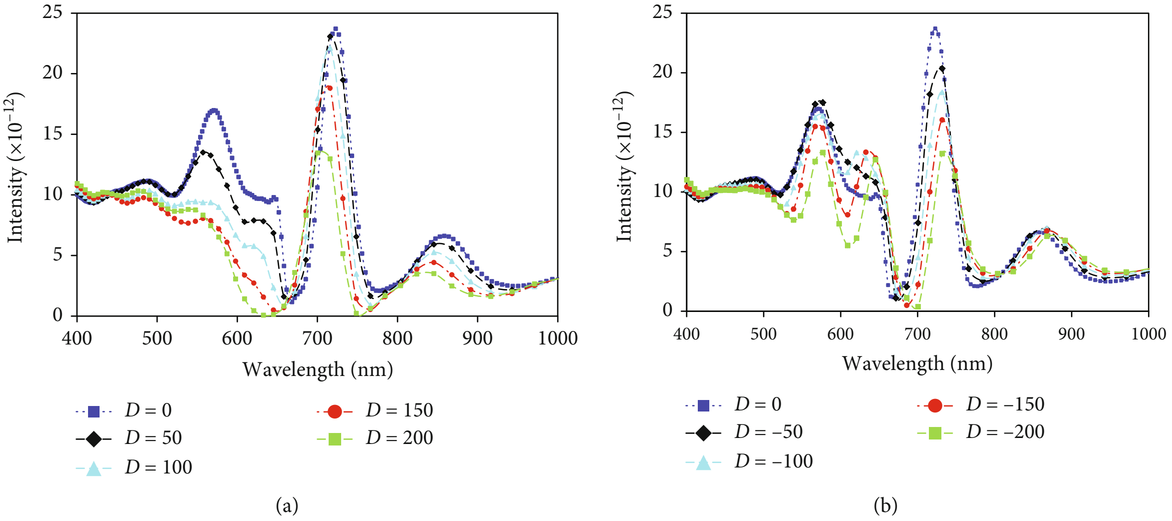


Figure 2: Modification of the relative position of the upper and lower concentric rings.



(a)

(b)

Figure 3: Transmission spectra in the far-field region at 0° collection angle for different values of D: (a) $D > 0$ and (b) $D < 0$.

was used to calculate the EM field intensity and distribution before and after the incident light hit the studied structures.

Based on the structure shown in Figures 1(a) and 1(b), some more geometric modifications were applied to the structure. The most exciting modification was found to be the impact of the relative position of the upper and lower concentric-ring grooves. As shown in Figure 2, the relative distance between the centers of the upper and lower adjacent concentric rings was defined as D, and 0 for the symmetric structures shown in Figure 1. For a positive value of D, the lower concentric-ring grooves remained closer to the center than the upper ones; while for a negative value of D, the opposite observation was recorded.

The transmission spectra shown in Figure 3 were obtained by varying the values of D within the range of -200 nm to +200 nm, while other parameters (film thickness, etching depth, concentric ring period, and the number of the concentric rings, etc.) remained unchanged. Since our structure is perfectly symmetrical, the optical wave simulation in this paper will not produce the phenomenon of polarization conversion.

Figure 3 reveals that when the relative distance D is greater than 0, the peak wavelength of the transmission spectrum experiences a blueshift. With the increase in the relative distance D, the magnitude of the blueshift increases, in addition to the decrease in peak field intensity. When D is less than 0, the peak wavelength of the transmission spectrum is redshifted. With the decrease in the relative position of D, the redshift magnitude increases, while the peak field intensity decreases. For further analysis of the relationship between the peak wavelength shift and D, the former is plotted as the y-axis against the latter as the x-axis in Figure 4. The relationship between the peak field intensity and D is plotted in Figure 5.

From Figure 4, it is obvious that the peak wavelength shift and the relative distance D have a near-linear relationship, and the fitting formula is in

$$\text{shift wavelength} = -0.0839 \times D - 0.2182. \qquad (1)$$

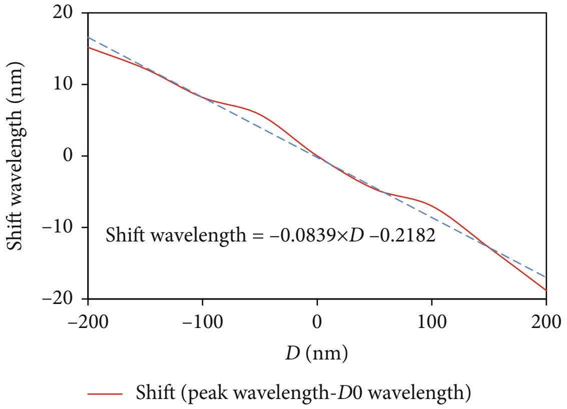


Figure 4: Relationship between the peak wavelength shift and D.

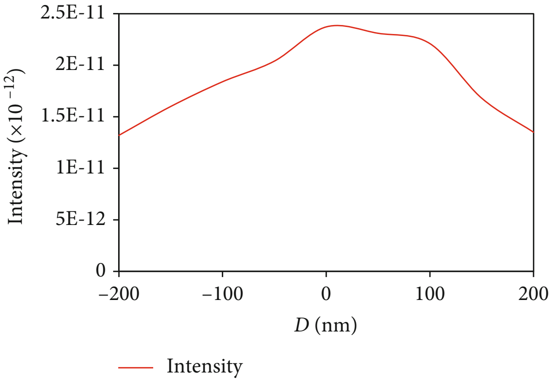


Figure 5: Relationship between the peak intensity and D.

The relationship curve is symmetric to the original point, which can be very useful and convenient in potential applications. It is well-known that both the upper and lower concentric-ring grooves can generate surface plasmonic waves. The final transmission peak arises as a result of the superposition and resonance of these waves. When the value of$D$is greater than 0, i.e., the lower grooves come closer to the

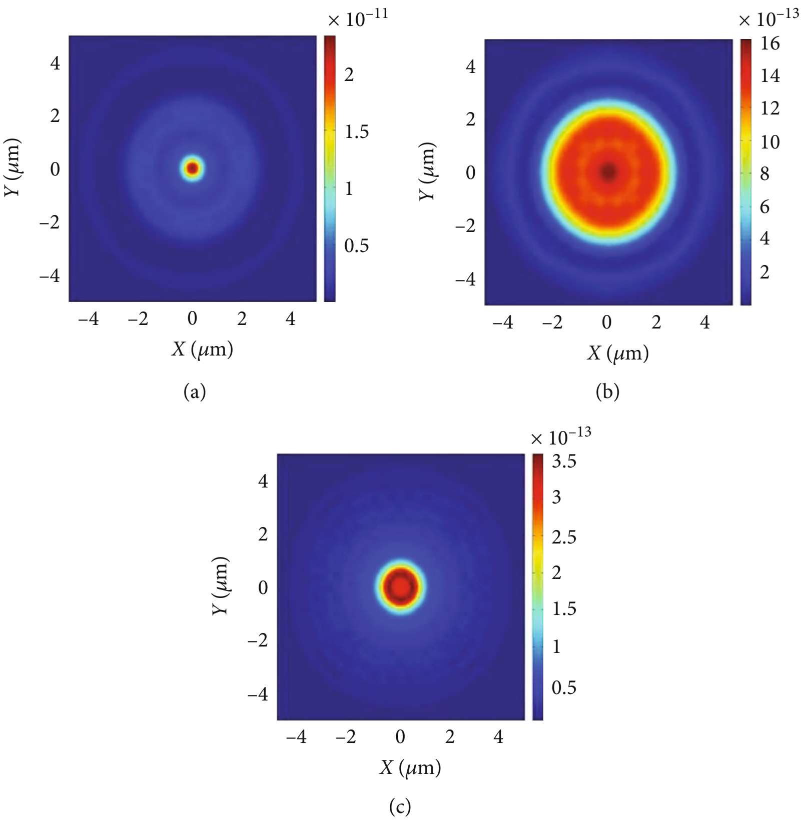


Figure 6: Far field of the film at 715 nm for (a) the proposed structure, (b) the reference aperture-only structure, and (c) the bull's eye structure.

center, this leads to the reduction in the effective aperture width of the lower structure. The aperture gets narrower for the entire structure, leading to a shorter peak wavelength, thus being blueshifted. Likewise, when the value of$D$is less than 0, i.e., the lower grooves tend to go further away from the center, and the effective aperture width of the lower structure gets wider for the entire structure, this leads to a longer peak wavelength, and hence is redshifted.

Figure 5 is easy to understand. The plasmonic wave produced by upper and lower concentric-ring grooves displays identical properties since the geometries and materials are all the same. When the upper and lower grooves get aligned ($D = 0$), the two surface plasmonic waves resonate. Thus, the highest transmission is generated. As a result, the peak transmission reaches a maximum when $D = 0$. The shift from the aligned position ($D > 0$ or $D < 0$) destroys the resonance and causes a decrease in the transmission intensity.

3. Details of the Interference Pattern Subheadings

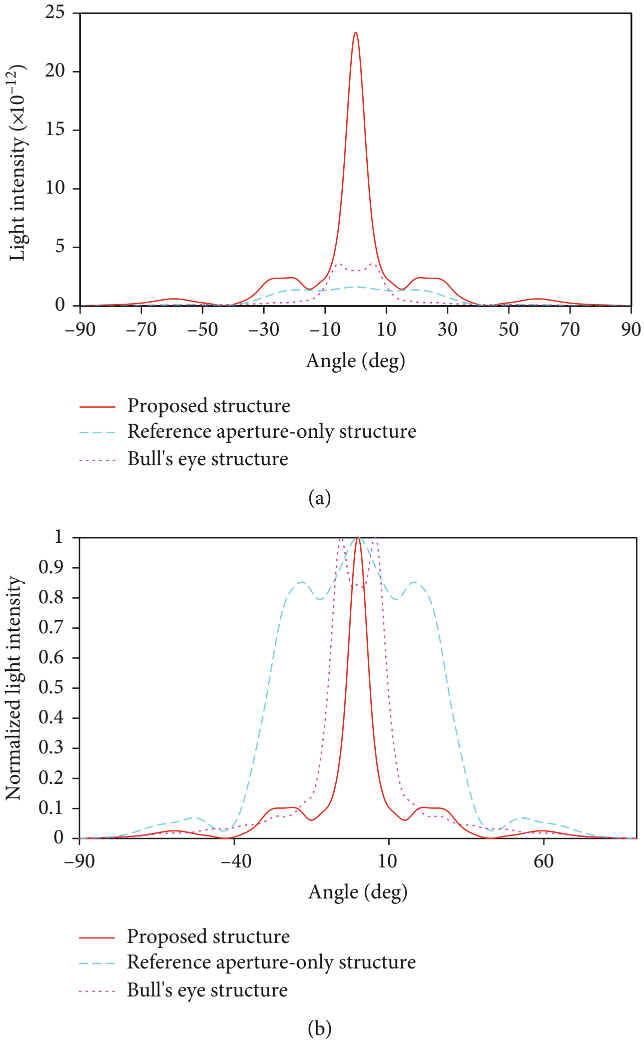
Since the transmission spectrum was modulated by a double-ring structure, a valid question arose about the process of change of the interference patterns for the proposed structure. The detailed interference patterns of the three structures shown in Figure 1 produced results based on which this question could be answered. The detailed discussion is presented in the following subsections.

3.1. Enhancement of the Interference Patterns. The far-field intensity patterns at 715 nm are shown in Figure 6. Figure 6(a) is the far-field pattern of the proposed structure, Figure 6(b) is the reference aperture-only structure, and Figure 6(c) is that of the bull's eye structure [4]. After adding a series of concentric rings in the surroundings, the intensity of the transmitted spectrum gets noticeably concentrated in the central area of the proposed structure, and the intensity at the center increases by one order of magnitude. Moreover, tightly squeezed circular interference fringes are produced by the proposed structure, as compared to the aperture-only structure. The proposed structure has the largest electric field intensity.

The corresponding angle dependence relationship at 715 nm can be a more precise description of the pattern squeeze, as shown in Figure 7.

Figure 7(a) displays the near 20-fold increase in the transmitted light intensity magnitude of the proposed structure versus the aperture-only structure. Figure 7(b) displays the normalized angle dependence, which clearly demonstrates that the distribution of the side lobes in the transmission pattern is practically the same for the two structures, and the crucial difference lies in the distribution of light intensity. The light intensity is highly concentrated in the main lobe for the proposed structure, which is nearly 10 times higher than the first side lobe pair. In contrast, the first side lobe pair is nearly 85% of the main lobe in the aperture-only structure.

This squeezed pattern is attributable to the strong resonance of the stimulated SPPs with the double-ring



(a)

(b)

FIGURE 7: The angle dependence of the transmission light intensity at 715 nm of the structure: original data (a) and normalized data (b).

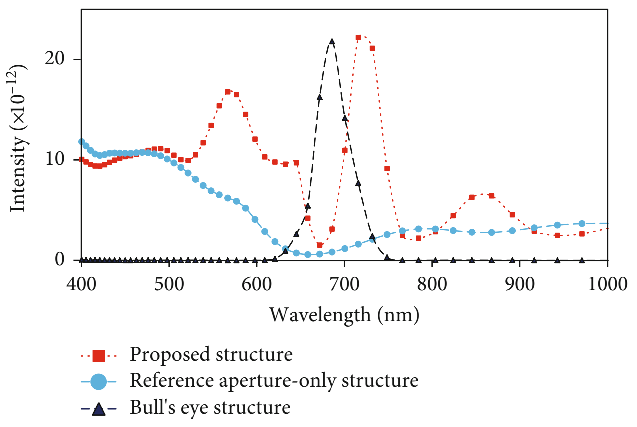


FIGURE 8: Transmission spectra for the three structures studied in the far-field region at 0° collection angle. Color code: red squares joined by red line: proposed structure; blue dots joined by blue line: aperture-only structure; purple triangles joined by purple line: bull's eye structure.

interference and can be useful in certain applications, including high-energy physics and coherent laser light focusing.

3.2. Enhancement of the Transmission Spectra. The transmission spectra of the three structures studied at a collection angle of 0° are shown in Figure 8.

From Figure 8, it can be clearly understood that in the aperture-only structure, the transmission spectrum curve is comparatively flat, with a few shallow ups and downs, but without any evident frequency-selectivity characteristics. While the bull's eye structure displays one clear transmission peak at 685 nm, transmission at other wavelengths is negligible. The proposed structure displays combined characteristics of the aperture-only structure and bull's eye structure, with signs of enhanced power transmission as well as stronger interference. Understanding of the transmission spectrum of the proposed structure is based on the following concepts: the double-ring aperture controls the interference patterns over the whole wavelength range, but the SPP effect takes place only at a specific SPP frequency, as evident from the transmission peak of the bull's eye structure. Therefore, supervision of the two structures, i.e., the proposed structure in this work, shows the fundamental interference patterns as the background, with strong modification by SPP around the SPP wavelength.

A more careful observation of the transmission spectra indicates that the total transmission power of the proposed structure is greater than the aperture-only structure, implying a much higher energy-collection capability. The peak intensity for the proposed structure is roughly twice of the aperture-only structure. This is because the SPP mode gets excited by the surrounding grooves that result in efficient collection of the surrounding energy.

The redshift of the transmission peak of the proposed structure from 685 nm to 715 nm, as compared to the bull's eye structure, is an interesting observation. This phenomenon most likely occurs because the concentric-ring in the center can become equivalent to the wider aperture in the bull's eye structure. As it is well-known, for bull's eye structures, the wider the central through-aperture is, the longer the peak wavelength will be.

4. Conclusions

In this paper, a new structure was proposed by combining the double-ring structure for interference with the bull's eye structure for SPP stimulation. It was observed that the new structure could introduce a stronger interference pattern and modify the transmission spectrum, which led to some interesting results. By altering the relative position of the upper and lower periodic grooves, both the intensity and the position of the peak of the transmission spectrum could be precisely adjusted. The results are useful in potential applications in both fundamental physics and applied optics.

Data Availability

The detailed parameter data of this article has been listed in the paper; according to this data, everyone can get the results of this paper.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

This work is supported by the Young Innovative Talents Project, Guangdong Province (No. 2019KQNCX069); Special Fund for Science and Technology Innovation Strategy of Guangdong Province (Climbing plan, pdjh2020b0344); Guangzhou Science and Technology Innovation Development Special Fund Project; and National Key R&D Program of China–Industrial Internet Application Demonstration-Sub-Topic Intelligent Network Operation and Security Protection (2018YFB1802402).

References

[1] J. Qiu, Z. Tian, C. Du, Q. Zuo, S. Su, and B. Fang, "A survey on access control in the age of internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 4682–4696, 2020.

[2] J. M. Yi, A. Cuche, E. Devaux, C. Genet, and T. W. Ebbesen, "Beaming visible light with a plasmonic aperture antenna," *ACS Photonics*, vol. 1, no. 4, pp. 365–370, 2014.

[3] P. Chen, C. Chen, S. Qin et al., "Efficient planar plasmonic directional launching of linearly polarized light in a catenary metasurface," *Physical Chemistry Chemical Physics*, vol. 22, no. 47, pp. 27554–27559, 2020.

[4] C. Cen, H. Lin, C. Liang et al., "Tunable plasmonic resonance absorption characteries-tics in periodic H-shaped graphene arrays," *Superlattices and Microstructures*, vol. 120, pp. 427–435, 2018.

[5] B. Eftekharinia, "Highly confined long range transmission of a surface plasmon polariton mode in a novel design of metallic slit-groove nanostructures," *Optik*, vol. 194, p. 163103, 2019.

[6] S. Basudeb, K. Roy, B. Yaara, and P. Yehiam, "Plasmonic flat surface Fabry-Perot interferometry," *Nanophotonics*, vol. 7, no. 3, pp. 635–641, 2018.

[7] B. Hauer, C. E. Marvinney, M. Lewin et al., "Exploiting phonon-resonant near-field interaction for the nanoscale investigation of extended defects," *Advanced Functional Materials*, vol. 30, no. 10, p. 1907357, 2020.

[8] B. Zhao and J. Yang, "New effects in an ultracompact Young's double nanoslit with plasmon hybridization," *New Journal of Physics*, vol. 15, no. 7, article 073024, 2013.

[9] G. Matan, S. Omer, W. Adam, A. Hannah, and G. David, "Second harmonic generation hotspot on a centrosymmetric smooth silver surface," *Light Science & Applications*, vol. 7, no. 1, article 49, 2018.

[10] H. Li, Y. Xu, G. Wang, T. Fu, L. Wang, and Z. Zhang, "Converting surface plasmon polaritons into spatial bending beams through graded dielectric rectangles over metal film," *Optics Communications*, vol. 383, pp. 423–429, 2017.

[11] G. Li and Q. Xiong, "Scattering by abrupt discontinuities on photonic nanowires: closed-form expressions for domain reduction," *Optics Express*, vol. 22, no. 21, pp. 25137–25148, 2014.

[12] G. Soavi, G. Wang, H. Rostami et al., "Broadband, electrically tunable third-harmonic generation in graphene," *Nature Nanotechnology*, vol. 13, no. 7, pp. 583–588, 2018.

[13] Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, "A distributed deep learning system for web attack detection on edge devices," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1963–1971, 2020.

[14] C. S. Perera, K. C. Vernon, A. M. Funston, H. Cheng, F. Eftekhari, and T. J. Davis, "Excitation of bound plasmons along nanoscale stripe waveguides: a comparison of end and grating coupling techniques," *Optics Express*, vol. 23, no. 8, pp. 10188–10197, 2015.

[15] H. Zhu, T. Xu, Z. Wang et al., "Flat metasurfaces to collimate electromagnetic waves with high efficiency," *Optics Express*, vol. 26, no. 22, pp. 28531–28543, 2018.

[16] M. Li, Y. Sun, H. Lu, S. Maharjan, and Z. Tian, "Deep reinforcement learning for partially observable data poisoning attack in crowdsensing systems," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6266–6278, 2020.

[17] J. Long, H. Yi, H. Li, Z. Lei, and T. Yang, "Reproducible ultrahigh SERS enhancement in single deterministic hotspots using nanosphere-plane antennas under radially polarized excitation," *Scientific Reports*, vol. 6, no. 1, article 33218, 2016.

[18] F. A. Koenderink, "Single-photon nanoantennas," *ACS Photonics*, vol. 4, no. 4, pp. 710–722, 2017.

[19] J. Bland-Hawthorn, M. Sellars, and J. Bartholomew, "Quantum memories and the double-slit experiment: implications for astronomical interferometry," 2021, https://arxiv.org/abs/2103.07590.

[20] J. Qi, T. Kaiser, A. E. Klein et al., "Enhancing resonances of optical nanoantennas by circular gratings," *Optics Express*, vol. 23, no. 11, pp. 14583–14595, 2015.

[21] L. Zheng, U. Zywietz, A. Evlyukhin, B. Roth, L. Overmeyer, and C. Reinhardt, "Experimental demonstration of surface plasmon polaritons reflection and transmission effects," *Sensors*, vol. 19, no. 21, p. 4633, 2019.

[22] Z. Li, Y. Sun, K. Wang et al., "Tuning the dispersion of effective surface plasmon polaritons with multilayer systems," *Optics Express*, vol. 26, no. 4, pp. 4686–4697, 2018.

[23] Y. P. Qi, P. Y. Zhou, X. W. Zhang, C. M. Yan, and X. X. Wang, "Enhanced optical transmission by exciting hybrid states of Tamm and surface plasmon polaritons in single slit with multi-pair groove nanostructure," *Acta Physica Sinica*, vol. 67, no. 10, article 107104, 2018.

[24] S. Wang, X. Wang, and Y. Zhang, "Simultaneous airy beam generation for both surface plasmon polaritons and transmitted wave based on metasurface," *Optics Express*, vol. 25, no. 20, pp. 23589–23596, 2017.

[25] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "CorrAUC: a malicious Bot-IoT traffic detection method in IoT network using machine-learning techniques," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3242–3254, 2021.

[26] T. Zhang, C. Wang, H. Chen et al., "Controlled multichannel surface plasmon polaritons transmission on atomic smooth silver triangular waveguide," *Advanced Optical Materials*, vol. 7, no. 21, p. 1900930, 2019.

[27] Z. Tian, X. Gao, S. Su, and J. Qiu, "Vcash: a novel reputation framework for identifying denial of traffic service in internet of connected vehicles," *IEEE Internet of Things Journal*, vol. 7, no. 5, pp. 3901–3909, 2020.

[28] H. Sun, Y. Zhu, B. Gao, P. Wang, and Y. Yu, "Polarization-dependent quasi-far-field superfocusing strategy of nanoring-based plasmonic lenses," *Nanoscale Research Letters*, vol. 12, no. 1, p. 386, 2017.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5569365, 13 pages
https://doi.org/10.1155/2021/5569365

Research Article

Securing NDN-Based Internet of Health Things through Cost-Effective Signcryption Scheme

Aroosa,[1] **Syed Sajid Ullah**,[2] **Saddam Hussain**,[2] **Roobaea Alroobaea**,[3] **and Ihsan Ali**[4]

[1]*Department of Computer Sciences, The Institute of Management Sciences, Lahore 54660, Pakistan*
[2]*IT Department, Hazara University, Mansehra 21120 KP, Pakistan*
[3]*Department of Computer Science, College of Computers and Information Technology, Taif University, P. O. Box 11099, Taif 21944, Saudi Arabia*
[4]*Department of Computer System and Technology, Faculty of Computer Science and Information Technology, University of Malaya, 50603 Kuala Lumpur, Malaysia*

Correspondence should be addressed to Ihsan Ali; ihsanalichd@siswa.um.edu.my

Received 7 February 2021; Revised 10 March 2021; Accepted 19 March 2021; Published 7 April 2021

Academic Editor: Habib Ullah Khan

The Internet of Health Things (IoHT) is an extended version of the Internet of Things that is acting a starring role in data sharing remotely. These remote data sources consist of physiological processes, such as treatment progress, patient monitoring, and consultation. The main purpose of IoHT platform is to intervene independently from geographically remote areas by providing low-cost preventive or active healthcare services. Several low-power biomedical sensors with limited computing capabilities provide IoHT's communication, integration, computation, and interoperability. However, IoHT transfers IoT data via IP-centric Internet, which has implications for security and privacy. To address this issue, in this paper, we suggest using named data networking (NDN), a future Internet model that is well suited for mobile patients and caregivers. As the IoHT contains a lot of personal information about a user's physical condition, which can be detrimental to users' finances and health if leaked, therefore, data protection is important in the IoHT. Experts and scholars have researched this area, but the reconstruction of existing schemes could be further improved. Also, doing computing-intensive tasks leads to slower response times, which further worsens the performance of IoHT. We are trying to resolve such an error, so a new NDN-based certificateless signcryption scheme is proposed for IoHT using the security hardness of the hyperelliptic curve cryptosystem. Security analysis and comparisons with existing schemes show the viability of the designed scheme. The final results confirm that the designed scheme provides better security with minimal computational and communicational resources. Finally, we validate the security of the designed scheme against man-in-the-middle attacks and replay attacks using the AVISPA tool.

1. Introduction

The Internet of Health Things (IoHT) refers to the collection of biomedical sensors and applications coupled with the networks as shown in Figure 1. Many healthcare providers use IoHT applications to improve treatments and patients' experience, reduce defects, control diseases, and reduce costs [1]. However, different healthcare things are introducing new aggressive approaches to healthcare infrastructure. This is attributed to the subsequent reasons:

(1) Medical things mainly transmit the sensitive data of patients

(2) Problems of incompatibility and complexity arise from the interaction of emerging devices and the various networks connected to them [2]

(3) As a growing sector, healthcare manufacturers are adopting IoT solutions regardless of safety. As a result, new security challenges related to confidentiality, authenticity, and availability

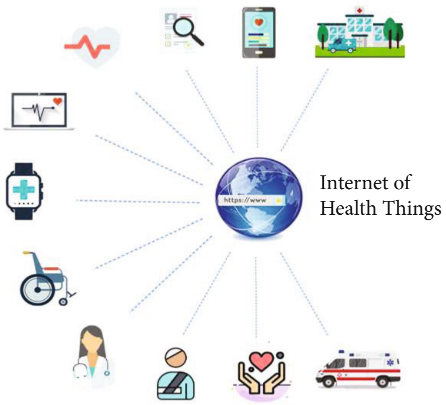


Figure 1: Internet of Health Things.

(4) As most IoT devices transmit and receive sensitive data wirelessly, this can put IoHT at risk for wireless sensor network security breaches [3]. Based on the IoHT environment's criticality, such security and privacy accidents can have devastating consequences such as loss of life and financial loss. In a healthcare information system, the details of patients can be preserved in the form of electronic health records accessible to medical specialists whenever the patient travels to the hospital. However, IoHT transfers data over the traditional Internet paradigm with risks associated with mobility and security. Therefore, IoHT risks need to be identified and assessed to provide better decision-making when adopting or constructing a secure and reliable IoHT scheme [4]

To tackle this, named data networking (NDN) is a best-chosen architecture of information-centric networking (ICN) [5]. The NDN application uses semantically meaningful, application-defined, and hierarchical names for the publication of data. Once the data is named and published, the user can send an interest packet to the network by specifying the requested content's name. Simultaneously, the intermediate NDN routers preserve a name-based forwarding table, which makes the longest prefix match in the name of interest and is sent through the appropriate interfaces [6]. Once the provider of the content receives the interest, it returns the signed data packet. The intermediate routers store these data packets in the content store for future requests. For more details regarding NDN, we refer the reader to some related publications [7, 8].

Traditionally, with a strong cryptographic scheme, malicious attacks can be prevented. Consequently, the cryptographic scheme must meet the security requirements of such as authentication, confidentiality, antireplay attack, integrity, and nonrepudiation [9]. On the other hand, the cryptographic perspectives that are used to secure the information are RSA-based cryptography [10, 11], symmetric key cryptography [12, 13], bilinear pairing [14, 15], elliptic curve cryptography (ECC) [16, 17], and hyperelliptic curve cryptography (HCC) [18, 19]. However, symmetric key-based schemes have major problems of key distribution. In contrast, RSA-based schemes incur high computational and communicational costs due to modular exponential complexities, bilinear pairing-based schemes suffer from heavy pairing operations, and ECC-based schemes outperform RSA. At the same time, HCC surpasses ECC in providing the same security features with lower cost complexities such as communication overhead, computation cost, and memory requirements.

HCC-based schemes require less storage and a smaller key of size of 80 bits in contrast to the 160 bits key of ECC and 1024 bits of RSA. They produce fewer ciphers compared to other public key cryptographic schemes. Because of these features, HCC is an attractive cryptographic phenomenon that provides security for systems utilizing limited computing resources. On the other hand, Zheng [20] introduced the concept of signcryption, which connects encryption and signature logically in a single step to reduce the cost complexities. Prior to the actual construction of signcryption, the encryption-than-signature was used to obtain privacy and authenticity. Zheng in his proposal showed that signcryption saves 50% of computation time and 85% of communicational costs as compared to the encryption-than-signature process. However, the proposed scheme of Zheng was constructed on public key cryptography (PKC) where the user's public key is a randomly selected string, so it requires a trusted entity such as certification authority (CA) to issue a certificate to link the user's public key with his/her corresponding identity. Unfortunately, the PKC suffers from the high cost of certificate management. This prevents the PKC from spreading to the real world. To reduce the burden of certificate management, Shamir [21] introduced an identity-based cryptography (IBC), where the identity of the particular users like IP address and telephone number can be used as his/her public key, thereby removing the certificate and simplifying key management. In an IBC, a reliable private key generator (PKG) is required to generate the user's private key, so the key escrow problem is inborn in the IBC.

Al-Riyami and Paterson [22] introduced certificateless cryptosystem (CLC), to eliminate key escrow problems encountered in IBS and certificate management issues in traditional PKC. In a CLC, trusted KGC is used to generate the partial private key for both the sender and receiver. The user must produce a secret value for himself/herself to combine the partial private key (PPK) with secret value to create a full private key. There has been a lot of focus on it since the introduction of CLC.

However, the first CLC scheme was presented by Barbosa and Farshim [23], combining the concepts of CLC and signcryption. Since then, many CLC schemes have been proposed in the literature [24–34].

1.1. Motivation and Contributions.

Security for all health fields is always a priority in modern communication technology. However, due to limited computing resources, the implementation of an efficient and appropriate security scheme for IoHT remains an ongoing challenge. The IoHT requires a security scheme that minimizes computing, communication, and storage overhead. Although the current complex cryptographic methods, i.e., ECC and bilinear pairing, are resulting in high-cost complexities, these cryptographic algorithms are

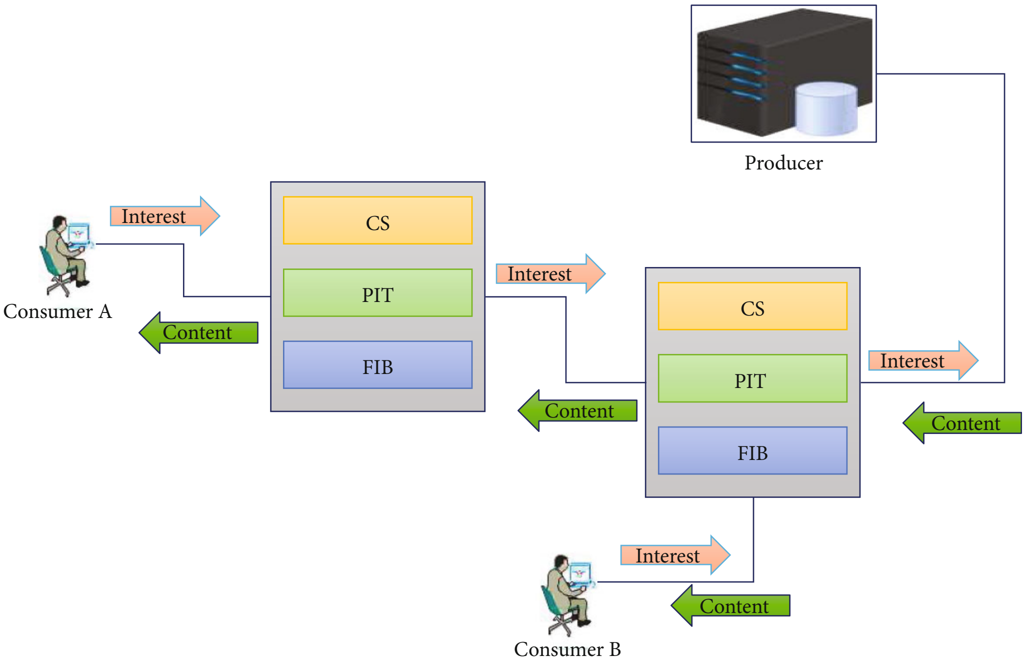


Figure 2: Basic NDN architecture with content distribution.

not compatible with low computing devices of IoHT systems. For creating a practical IoHT solution that requires minimal computation, it is necessary to use HCC. As the HCC offers the same level of security utilizing smaller key sizes in contrast to ECC and bilinear pairing, we describe our major contribution below.

(i) We designed an NDN-based IoHT scheme using the security hardness of HCC

(ii) The designed scheme offers the security services of confidentiality, authenticity, integrity, unforgeability, and nonrepudiation

(iii) Security analysis and comparisons with existing schemes show the designed scheme's viability. The obtained results confirm that the designed scheme provides better security with minimal computational and communicational resources

(iv) We validate the security of the designed scheme using the AVISPA tool

(v) Finally, we deployed the newly designed scheme on NDN-based IoHT

1.2. Overview of NDN. NDN is a future network architecture designed to cover IoT users' demands such as efficient content distribution, improved mobility, and scalable connectivity to the end-users [5]. NDN is designed to offer in-network caching and named-based routing that eliminates the location dependency, connectivity, and content distribution problems of IP-based Internet. Moreover, NDN supports 2 types of packets, namely, interest packet (IP) and data packet (DP). The IP consists of the requested content by name, while the DP consists of the requested content with information about the provider. Moreover, each NDN node maintains 3 kinds of data structures [6] as shown in Figure 2. The CS is used as a local cache memory that stores the copies of passing contents for future use to facilitate the end-users. The PIT is used as an entry list that keeps the records of incoming/outgoing IP and DP. The FIB forwards the IP and DP from one node to another using traditional protocols such as OSPF and BGP [7].

1.3. Road Map of the Article. The rest of the paper is structured as follows: Section 2 provides the knowledge about the existing literature, Section 3 provides the preliminaries of HCC, Section 4 presents the construction and the proposed network model, Section 5 provides the security analysis, Section 6 delivers the performance and discussion with the existing scheme in terms of cost complexities, and Section 7 contains the overall deployment of the designed scheme on IoHT. Finally, Section 8 contains the conclusion and implementation of the designed scheme using AVISPA tool.

2. Preliminaries

2.1. Hyperelliptic Curve Discrete Logarithm Problem (HCDLP)

(i) Let $\Theta \in \{1, 2, 3, 4, 5, \cdots, n-1\}$and $\mathscr{B} = \Theta \cdot D$, then finding Θ and $\mathscr{B}$ is known as HCDLP

2.2. Hyperelliptic Curve Computational Diffie-Hellman Problem (HCCDHP)

(i) Let $\Theta, \Omega \in \{1, 2, 3, 4, 5, \cdots, n-1\}$ and $\mathscr{B} = \Theta \cdot D, \delta = \Omega \cdot \Theta \cdot D$, then finding Θ and Ω from $\mathscr{B}$ and δ is known as HCCDHP

2.3. Syntax for the Design Scheme. The proposed scheme for NDN-based Internet of Health Things consists the following algorithms:

(i) Setup: the network manager (NM) picks a parameter of security (ℓ) as input and generates the master secret key ($\mathscr{M}$), master public key ($\mathscr{M}_{\text{pub}}$), and public parameter set ($\mathscr{P}$)

(ii) Generation of partial private key: the NM takes the users' identities (ID_u), $\mathscr{M}$, $\mathscr{M}_{\text{pub}}$, and $\mathscr{P}$ as in input and generates the partial private keys ($\mathscr{X}_u$) for users

(iii) Secret value setting: the users pick a private number as a secret value ($\mathscr{S}$)

(iv) Private key generation: the users generate private keys (PV_K) by taking $\mathscr{S}$ and $\mathscr{X}_u$ as input

(v) Public key generation: the users generate their public keys PK_u by taking $\mathscr{S}$ and deviser of HCC ($\mathscr{D}$) as input

(vi) Signcryption: the provider of the content will produce the signcrypted content (δ) by taking as input ID_u, PK_u,$\mathscr{S}$, $\mathscr{P}$, and $\mathscr{X}_u$

(vii) Unsigncryption: the consumer of the content will unsigncrypt the received δ. For this process, it takes as input δ, ID_u, PK_u,$\mathscr{S}$, $\mathscr{P}$, and $\mathscr{X}_u$

2.4. Threat Model. In the designed scheme, we adopt the most popular treat model, i.e., Dolev-Yao [35]. According to this model, the communication among two or more than two entities is not secure and trusted because the attacker has full instructions to expose the signcrypted content and to forge the signature. There are several security threats in the NDN-based IoHT environment. It means that a user can edit or delete strategic information from competitors. To preserve the security and authenticity of NDN-based IoHT devices, authentic and secure communication between entities is required.

For the security explanation of the designed scheme, we take two types of adversaries, i.e., (type I and type II) [30, 36].

(i) Type I adversary: type I adversary is often considered to be an external attacker who does not have the master secret key and can request a user's public key and replace it with its own chosen value

(ii) Type II adversary: type II adversary is also considered as malicious KGC that can compute a user's PPK using the master secret key; however, this type of adversary is not able to replace the public key of the users

3. Related Work

The related work consists of two parts like IoHT schemes in NDN and certificateless signcryption approaches.

3.1. IoHT Schemes in NDN. Saxena et al. [37] in 2015 presented a healthcare scheme for NDN. The given scheme was the first solution for NDN-based healthcare. Two years later [38], Saxena and Raychoudhury proposed another healthcare scheme in the NDN network. The goal of the scheme is to provide authenticity for emergency messages. Unfortunately, both schemes did not provide security for NDN-based healthcare. Wang and Cai [39] designed a framework to secure healthcare in NDN-enabled edge cloud. It was the first security framework of healthcare NDN. The author highlights the positive aspects of NDN for the enhancement and efficiency of healthcare. However, the authors used weighty pairing operations of bilinear pairing in attribute-based encryption.

3.2. Certificateless Signcryption Schemes. Nowadays, data transmission through the Internet is a famous communication technique because of which security becomes a major issue of concern. To save the personal information of the users and avoid unauthorized access to data, we must ensure authentication, confidentiality, and integrity of data [24]. To overcome confidentiality, the encryption method is in use. Simultaneously, for integrity, authentication, and nonrepudiation, the digital signature is operative. In the previous era, the sender uses to encrypt and then sign the document before sending it to the receiver which is known as the sign-then-encrypt method. But the sign-then-encrypt method has a flaw as it is a time-consuming process and the system needs more power which intrudes the system efficiency.

To remove the KEP, Barbosa and Farshim [23] together introduce a CLC signcryption method by achieving the CLC encryption and signature in one step. Zhou et al. [25] presented a new CLC signcryption and proved the security of their scheme based on security according to Diffie-Hellman problem. Later on, efficient CLC signcryption based on the standard scheme was introduced by Rastegari and Berenjkoub [26]. Their work shows their scheme is safer and more effective than all existing oracle model-based CLC signcryption schemes. Without BP, in 2017, Yu and Yang [27] come up with a new CLC signcryption scheme and proved the security in ROM.

Further, according to Yu and Yang, the proposed algorithm is suitable for applications like an email system and online sale. Zhou [28] proposed a new CLC signcryption technique. The security of the scheme is based on BP using the standard model. Based on the efficiency and the hardiness of the elliptic curve discrete logarithm problem (ECDLP), for the best solutions of cloud storage, Luo and Ma [29] introduced a CLC hybrid signcryption technique. However, the given scheme was constructed on the security hardness of the ECDLP, which is not suitable for the IoT environment. In 2020, Liu et al. [30] proposed a scheme for access control in WBS networks by using CLC signcryption. The design scheme is based on RSA. The security proved under the ROM. In the same year, Kasyoka et al. [31] found out the security shortcoming and provided an improved CLC signcryption scheme.

In 2017, Li et al. [32] constructed a CLC signcryption with access control for industrial wireless sensor networks.

According to the authors, the given scheme achieves the additional security properties of ciphertext authenticity, insider security, and public verifiability. Though the given scheme is more efficient than the previous access control schemes, however, the scheme of Li et al. was based on bilinear pairing which makes it inefficient for the resource-constrained environment of IoT due to heavy pairing costs.

In late 2020, Swapna et al. [34] presented a new CLC signcryption scheme under the security hardness of ECDLP under ROM. According to the authors, the proposed scheme achieves the additional security requirement of public verifiability with strong security against various types of malicious adversaries. Unfortunately, the scheme of Swapna et al. was constructed on the concept of ECC, which utilizes 160 bits keys for providing security, which is still not affordable for the limited resource devices.

However, all the above [24–34] schemes suffer from heavy communication and computation costs due to using heavy bilinear pairing, RSA, and ECC.

4. Proposed Scheme for NDN-Based Internet of Health Things

4.1. Proposed Network Model for NDN-Based Internet of Health Things. Figure 3 shows our proposed network model for NDN-based Internet of Health Things. The suggested model consists of the following entities and their functions.

(i) Network manager (NM): the NM is an authentic authority that manages and ensures secure data transformation among IoHT devices or users (consumer, producer)

(ii) Consumer: any IoHT device (such as smartphone and body sensor) or user (such as hospital, patient, and doctor) that are interested in IoHT data (like patient record stats and monitoring) in a secure way

(iii) Producer: any IoHT device (such as smartphone and body sensor) or user (such as hospital, patient, and doctor) that provides IoHT data (like patient record stats and monitoring) in a secure way

(iv) NDN nodes: the NDN nodes transfer IoHT interest and data/content between consumer and producer using NDN routing policy

In the suggested NDN-based IoHT model, at the start of communication, the IoHT devices or users need to be registered with NM. For this process, users or devices send their own identities to NM. Upon receiving, the NM generates a partial private key for users and sends it. Then, users use the partial private key and make their own public and private keys.

Let a consumer show interest in some healthcare-related data/content, then the NDN nodes will forward the interest to the producer using the traditional routing protocols such as OSPF and EIGRP. After receiving the interest, the producer will simply signcrypt the data/content and send it to the interested consumer using the reverse path. After the reception, the NDN node will forward the signcrypted data/-content through FIB. However, none of the NDN nodes will cache the content/data as it is signcrypted for a particular receiver. This process will repeat until the particular receiver user receives the interesting content/data. Here, we focus on the confidentiality and authenticity of NDN-based IoHT data, so the copy data/content will not be cached in any NDN node. Also, the data/content can only be verified with the private key of interested consumers to not facilitate any user if the content/data is cached in the intermediate NDN nodes.

4.2. Proposed Algorithm. The proposed algorithm comprises seven steps as described below. The symbols used in the construction of the designed scheme are mentioned in Table 1.

4.2.1. Setup. Given the security parameter (ℓ), this algorithm generates a master secret key ($\mathscr{M}$) and public parameter set ($\mathscr{P}$). The given algorithm is executed by the KGC and performs the following tasks.

(i) Select ($\mathscr{D}$) as a devisor of HCC of order q

(ii) Select a prime number $\mathscr{M}$, where $\mathscr{M} \in \preceq 1 \preceq (q-1)$

(iii) Compute $\mathscr{M}_{\text{pub}} = \mathscr{M} \cdot \mathscr{D}$ as his master public key

(iv) Select one-way hash functions of $(\text{SHA}-512) = H_1, H_2, H_3, H_4$

(v) The given algorithm keeps the master secret key with itself and advertises the public parameter set $\mathscr{P} = \{\mathscr{M}_{\text{pub}}, \mathscr{D}, q, H_1, H_2, H_3, H_4\}$ in the network

4.2.2. Set Secret Value. The given algorithm is executed on the participant's side (i.e., client and producer) with identity ID_u the participants select a random number from $S \in \preceq 1 \preceq (q-1)$ as a secret value and compute their public keys as $\text{PK}_u = \mathscr{S} \cdot \mathscr{D}$.

4.2.3. Partial Private Key Generation. The KGC executes the given algorithm. It selects a random number $R_n \in \preceq 1 \preceq (q-1)$ and computes $\mathscr{N}_u = R_n \cdot \mathscr{D}$. The KGC then computes the partial private key as follows:

(i) Compute $h_1 = H_1(\text{ID}_u, \mathscr{N}_u, \text{PK}_u, G\mathcal{O})\, \mathscr{M}_{\text{pub}}$

(ii) Compute $\mathscr{X}_u = R_n + \mathscr{M} \cdot h_1 \mod q$

(iii) Compute $\mathscr{T}_u = \mathscr{N}_u + H_1(\text{ID}_u, \mathscr{N}_u, \text{PK}_u, G\mathcal{O})\, \mathscr{M}_{\text{pub}}$

The KGC then forwards $(\mathscr{X}_u, \mathscr{T}_u, \mathscr{N}_u)$ to the participants using a private channel. The participants, upon receiving the $\mathscr{X}_u$, can verify the validity by checking $\mathscr{X}_u \cdot \mathscr{D} = \mathscr{N}_u + H_1(\text{ID}_u, \mathscr{N}_u, \text{PK}_u)\, \mathscr{M}_{\text{pub}}$.

4.2.4. Private Key Generation. In this algorithm, the participants set their private key as $\text{PV}_K = (\mathscr{X}_u, \mathscr{S})$

4.2.5. Signcryption. This given algorithm is performed on the producer side. For signcryption, the producer performs the following operations:

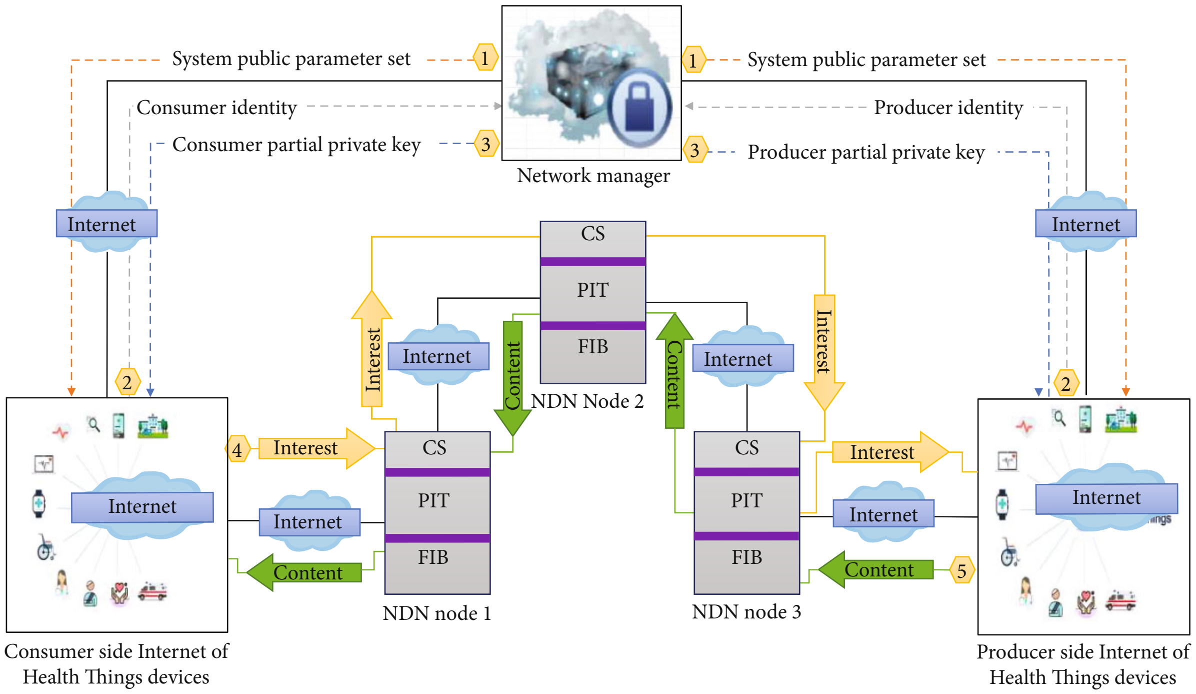


Figure 3: Proposed network model for NDN-based Internet of Health Things.

Table 1: List of notations.

S/no	Notations	Explanation
1	ℓ	Parameter of security with 80 bits size
2	$\mathcal{P}$	Public parameter set
3	H_1, H_2, H_3, H_4	Hash functions
4	ID_u	User identities
5	$\mathcal{M}$	Master secret key
6	$\mathcal{S}$	Secret values
7	$\mathcal{M}_{\mathrm{pub}}$	Master public key
8	$\mathcal{X}_u$	User partial private key
9	PV_K	User private key
10	$\mathcal{G}$	Fresh nonce
11	δ	Signcrypted content

Table 2: Software and hardware details.

System	Specification
Library	Multiprecision integer and rational arithmetic C
Operating system	Windows 7-64 bits
CPU	Intel Core i7-4510
RAM	8 GB

Table 3: Running time of major operations in computation complexity.

HECDM	SPMEC	E	BP	PBPM
0.48	0.97	1.25	14.90	4.31

(1) Pick a random number $\in \precsim 1 \precsim (q-1)$

(i) Compute $\mathcal{J} = \mathcal{W} \cdot \mathcal{D}$

(ii) Compute $\mathcal{Z}_c = \mathcal{W}_p \cdot \mathcal{T}_c$

(iii) Compute $\hbar_2 = H_2(\mathcal{J}, \mathcal{Z}_p, \mathcal{G}, \mathrm{ID}_c, \mathrm{PK}_p)$

(iv) Compute $\hbar_3 = H_3(\mathcal{J}, \mathcal{Z}_p, \mathcal{G}, \mathrm{ID}_c, \mathrm{PK}_c)$

(v) Compute $\mathcal{B}_p = \mathcal{X}_p + \mathcal{W}_p \cdot \hbar_2 + \mathcal{S}_p \cdot \hbar_3 \mod q$

(vi) Compute $\hbar_4 = H_4(\mathcal{Z}_p, \mathcal{J}, \mathcal{T}_c, \mathrm{ID}_c)$

(2) Produce a signcrypted text as $\delta = (\mathcal{J}, \mathcal{B}_p, \hbar_4)$ and send it to the consumer

4.2.6. Unsigncryption. With δ, ID_p, and $(\mathcal{T}_p, \mathrm{PK}_p)$, the decryption is as follows:

(i) Compute $\mathcal{Z}_p = \mathcal{X}_p \cdot \mathcal{T}_p$

(ii) Compute $\hbar_2 = H_2(\mathcal{J}, \mathcal{Z}_p, \mathcal{G}, \mathrm{ID}_c, \mathrm{PK}_p)$

(iii) Compute $\hbar_3 = H_3(\mathcal{J}, \mathcal{Z}_p, \mathcal{G}, \mathrm{ID}_c, \mathrm{PK}_c)$; if $\mathcal{B}_p \mathcal{D} = \mathcal{T}_p + \hbar_2 \cdot \mathcal{J} + \hbar_3 \cdot \mathrm{PK}_p$ holds, then the received signature is valid otherwise forged.

$$\hbar_4 = H_4(\mathcal{X}_c . \mathcal{J}, \mathcal{T}_c, \mathrm{ID}_c). \tag{1}$$

Table 4: Computation cost in terms of costly mathematical operations.

Schemes	Signcryption	Unsigncryption	Total cost in (ms)
[32]	$3\,\mathbb{E}$	$3\,\mathbb{E} + 1\mathbb{BP} + 1\mathbb{PBPM}$	$6\,\mathbb{E} + 1\,\mathbb{BP} + 1\mathbb{PBPM}$
[34]	$3\,\mathbb{SPMEC}$	$4\,\mathbb{SPMEC}$	$7\,\mathbb{SPMEC}$
[29]	$4\,\mathbb{SPMEC}$	$4\,\mathbb{SPMEC}$	$4\,\mathbb{SPMEC}$
Proposed	$4\,\mathbb{HECDM}$	$3\mathbb{HECDM}$	$7\,\mathbb{HECDM}$

Table 5: Computation cost analysis in milliseconds.

Schemes	Signcryption	Unsigncryption	Total cost
[32]	3.75	22.96	26.71
[34]	2.91	3.88	6.79
[29]	3.88	3.88	7.76
Ours	1.92	1.44	3.36

4.2.7. Consistency.

$$\begin{aligned} \mathscr{Z}_c &= \mathscr{W}\mathscr{T}_c = \mathscr{W}\left(\mathscr{N}_c + \hbar_1 \mathscr{M}_{\text{pub}}\right), \\ \mathscr{Z}_c &= \mathscr{X}_c \cdot \mathscr{J} = \mathscr{W}\,\mathscr{D}(R_n + \mathscr{M}\hbar_1) = \mathscr{W}\left(\mathscr{N}_c + \hbar_1 \mathscr{M}_{\text{pub}}\right). \end{aligned} \tag{2}$$

5. Security Analysis

Here, we provide a detailed analysis of the designed scheme for the security aspects of confidentiality, integrity, authentication, nonrepudiation, and unforgeability. Each of these aspects is discussed in more detail in the following sections.

5.1. Theorem (Confidentiality). A certificateless signcryption scheme is known to accomplish the security requirement of confidentiality if there is no possible adversary that can attain the provider's encryption key.

Proof. The designed scheme certifies the requirement of confidentiality if the adversary desires to obtain the content from signcrypted text (δ) where $\delta = (\mathscr{J}, \mathscr{B}_p, \hbar_4)$. In this case, he/she must have to find $\mathscr{J}$, $\mathscr{B}_p$, and $\hbar_4$. To fix $\mathscr{J}$, the adversary also needs to find $\mathscr{W}$ from $\mathscr{J} = \mathscr{W} \cdot \mathscr{D}$ which is infeasible due to the use of HCDLP. Furthermore, the adversary needs to calculate $\mathscr{B}_p$ here for $\mathscr{B}_p$ adversary should calculate $\mathscr{X}_p$ and $\mathscr{S}_p$ from $\mathscr{B}_p = \mathscr{X}_{p+}\mathscr{W}_p \cdot \hbar_2 + \mathscr{S}_p \cdot \hbar_3 \bmod q$ which is infeasible due to the use of HCDLP.

5.2. Theorem (Authentication). A certificateless signcryption scheme is known to accomplish the security requirement of authenticity if the content receiver is somehow able to verify the original source of content.

Proof. A client can use ID_p and $(\mathscr{T}_p, \text{PK}_p)$ to verify the signature from δ. Here, to generate δ in the producer side, the producer uses $(\mathscr{X}_p)$ and $(\mathscr{S}_p)$ which are equal to the private key of the producer of the content. Hence, the content is signed with the private key of the provider. So, the receiver of the content/message can easily verify the respective producer's identity to check the authenticity.

5.3. Theorem (Integrity). A certificateless signcryption approach is known to attain the security requirement of integrity if there is no possible adversary that can produce the equivalent hash value for the different sizes of the message.

Proof. The producer of the content/message takes the "hash value" "$\mathscr{B}_p = \mathscr{X}_{p+}\mathscr{W}_p \cdot \hbar_2 + \mathscr{S}_p \cdot \hbar_3 \bmod q$" before sending the content/message to the consumer. Suppose an adversary tries to change the cipher content, in that case, the content receiver can verify the ciphertext by doing the subsequent steps. The consumer of the content/message first computes $\mathscr{B}_p\mathscr{D} = \mathscr{T}_p + \hbar_2 \cdot \mathscr{J} + \hbar_3 \cdot \text{PK}_p$ and $\hbar_4 = H_4(\mathscr{X}_c \cdot \mathscr{J}, \mathscr{T}_c, \text{ID}_c)$; if it holds, then the content is valid; else, the content/-message has been changed.

5.4. Theorem (Unforgeability). Suppose an adversary is able to negotiate the $\mathscr{X}_p)$ of the provider, in that case, the certificateless signcryption approach meets the security requirement of unforgeability.

Proof. In the designed approach, if an adversary attempts to produce a legal signature, they need to compute $\mathscr{B}_p$ from $\delta = (\mathscr{J}, \mathscr{B}_p, \hbar_4)$, and for doing that, he/she needs to find $\mathscr{J}$. To fix $\mathscr{J}$, the adversary needs to obtain $\mathscr{W}$ from $\mathscr{J} = \mathscr{W} \cdot \mathscr{D}$, which is infeasible due to the security hardness of HCDLP.

5.5. Theorem (Nonrepudiation). A certificateless signcryption scheme is known to accomplish the security requirement of nonrepudiation if a producer/provider of the content/message cannot deny from his/her generated signcrypted ciphertext.

Proof. The content/message is normally signed with the private key $(\mathscr{X}_p)$ and $(\mathscr{S}_p)$ of the producer/provider. In the designed scheme, the consumer/receiver of the content/message can authenticate the identity ID_p of the provider. So, the provider of the content/message later cannot repudiate his own signature.

6. Complexity Analysis

We compared our scheme with previously suggested certificateless signcryption schemes on the following two bases.

6.1. Computation Cost. In the following section, we will demonstrate the computational cost complexity of our scheme

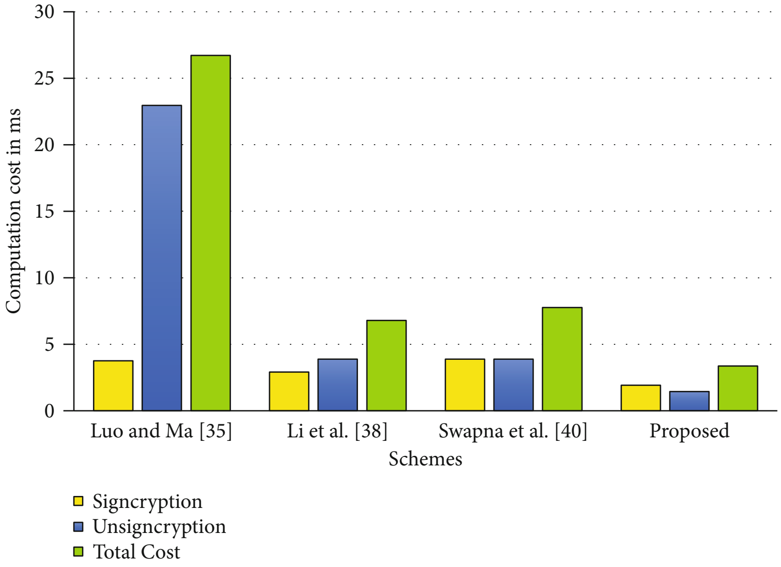


Figure 4: Computation cost complexity.

Table 6: Variables used in our analysis.

Name	Notation	Size (bits)
BP	$(\mathbb{G})$	1024
HCC	$(\mathbb{q})$	80
ECC	$(\mathbb{n})$	160
Message	$(\mathbb{m})$	512

Table 7: Communication overhead analysis in bits.

Schemes	Ciphertext size	Size (bits)
Luo and Ma [29]	$3(\mathbb{n}) + (\mathbb{m})$	992
Li et al. [32]	$3(\mathbb{G}) + (\mathbb{m})$	3584
Swapna et al. [34]	$3(\mathbb{n}) + (\mathbb{m})$	992
Ours	$3(\mathbb{q}) + (\mathbb{m})$	752

and previously suggested certificateless signcryption schemes such as Li et al. [32], Swapna et al. [34], and Luo and Ma [29]. To estimate the computational complexity, we considered the cost of signcryption and unsigncryption. However, to calculate the operational computation cost of any scheme, we mostly consider the costly mathematical operation used in that particular cryptographic scheme. For our computation complexity analysis, we take bilinear pairing ($\mathbb{BP}$), pairing-based point multiplication ($\mathbb{PBPM}$), exponential ($\mathbb{E}$), scalar point multiplication of elliptic curve ($\mathbb{SPMEC}$), and hyperelliptic curve devisor multiplication ($\mathbb{HECDM}$), respectively. The software and hardware specification [19, 33] used with the running time is shown in Tables 2 and 3.

From the results in Tables 4 and 5 and Figure 4, it is clear that our scheme works more efficiently in terms of computational complexity than the previous ones.

6.1.1. Cost Reduction. The cost reduction/percentage improvement can be attained using the given formula [18].

$$= \left(\frac{\text{Computation cost of previous scheme} - \text{Computation cost of designed scheme}}{\text{Computation cost of previous scheme}}\right) * 100. \tag{3}$$

(i) The percentage improvement of the designed scheme from Li et al. [32] is as follows:

$$= \left(\frac{26.71 - 3.36}{26.71}\right) * 100 = 87.42\%. \tag{4}$$

(ii) The percentage improvement of the designed scheme from Swapna et al. [34] is as follows:

$$= \left(\frac{6.79 - 3.36}{6.79}\right) * 100 = 50.51\%. \tag{5}$$

(iii) The percentage improvement of the designed scheme from Luo and Ma [29] is as follows:

$$= \left(\frac{7.76 - 3.36}{7.76}\right) * 100 = 56.70\%. \tag{6}$$

6.2. Communication Overhead. Here, we show a comparative analysis of the given scheme with the relevant existing schemes [29, 32, 34]. However, to calculate the operational communication overhead of any scheme, we mostly study the additional bits that an original message will carry. For our scheme, we used variables such as elliptic curve cryptosystem (ECC): $(\mathbb{n})$, hyperelliptic curve cryptosystem (HCC):

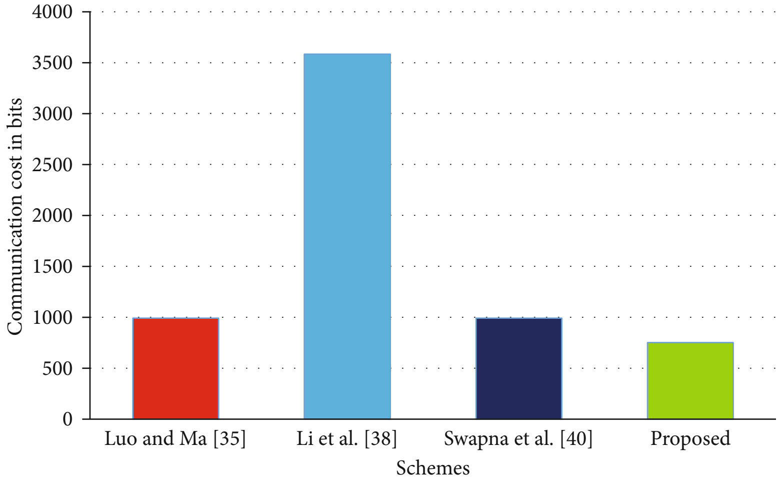


FIGURE 5: Communication cost complexity.

$(\mathbb{q})$, bilinear pairing (BP): $(\mathbb{G})$, and message: $(\mathbb{m})$ as further shown in Table 6.

From the results in Tables 7 and Figure 5, it is clear that our scheme works more efficiently in terms of communicational overhead than the previous ones.

6.2.1. Cost Reduction. The cost reduction/percentage improvement can be attained using the given formula [18].

$$= \left(\frac{\text{Computation cost of previous scheme} - \text{Computation cost of designed scheme}}{\text{Computation cost of previous scheme}}\right) * 100. \tag{7}$$

(i) The percentage improvement of the designed scheme from Li et al. [32] is as follows:

$$= \left(\frac{3584 - 752}{3584}\right) * 100 = 79.01\%. \tag{8}$$

(ii) The percentage improvement of the designed scheme from Swapna et al. [34] and Luo and Ma [29] is as follows:

$$= \left(\frac{6.79 - 3.36}{6.79}\right) * 100 = 50.51\%. \tag{9}$$

7. Deployment on NDN-Based Internet of Healthcare

Figure 6 shows a robust and secure deployment of the given scheme on the NDN-based Internet of IoHT. We consider many connected IoH devices that can exchange healthcare information for this deployment. Furthermore, the medical devices are linked to NDN policy [7, 8]. The complete deployment for secure communication is described in the subsequent steps.

7.1. Registration and Key Generation. In this phase, the KGC enrolls both the participants with itself. To do so, the KGC picks a security parameter (ℓ), selects D of HCC of order q, selects a prime number $\mathscr{M}$, where $\mathscr{M} \in \preceq 1 \preceq (q-1)$, as a master secret key, then computes $\mathscr{M}_{\text{pub}} = \mathscr{M} \cdot \mathscr{D}$, and selects one-way hash functions H_1, H_2, H_3, H_4. The KGC keeps the master secret key with itself and advertises the public parameter set $\mathscr{P} = \{\mathscr{M}_{\text{pub}}, \mathscr{D}, q, H_1, H_2, H_3, H_4\}$ in the network. After the advertisement of KGC, the consumer and producer first select random number from $\mathscr{S} \in \preceq 1 \preceq (q-1)$ as a secret value and compute their public keys as $\text{PK}_c = \mathscr{S} \cdot \mathscr{D}$ and $\text{PK}_p = \mathscr{S} \cdot \mathscr{D}$. Then, the participants send their identities $(\text{ID}_c, \text{ID}_p)$ to KGC. It selects a random number $R_n \in \preceq 1 \preceq (q-1)$ and compute $\mathscr{N}_u = R_n \cdot \mathscr{D}$. The KGC then computes the partial private key as compute $h_1 = H_1(\text{ID}_u, \mathscr{N}_u, \text{PK}_u)\, \mathscr{M}_{\text{pub}}$, compute $\mathscr{X}_u = R_n + \mathscr{M} \cdot h_1 \bmod q$, and compute $\mathscr{T}_u = \mathscr{N}_u + H_1(\text{ID}_u, \mathscr{N}_u, \text{PK}_u)\, \mathscr{M}_{\text{pub}}$. The KGC then forwards the $(\mathscr{X}_u, \mathscr{T}_u, \mathscr{N}_u)$ to the client/consumer/receiver of the content and provider of the content through a private channel. The consumer and producer upon receiving the $\mathscr{X}_u$ can verify the validity by checking $\mathscr{X}_u \cdot \mathscr{D} = \mathscr{N}_u + H_1(\text{ID}_u, \mathscr{N}_u, \text{PK}_u)\, \mathscr{M}_{\text{pub}}$.

7.2. Signcryption. Whenever a consumer of the content shows an interest in some healthcare information, after receiving, the producer will generate signcrypted content for the consumer as to pick a random number $\mathscr{W} \in \preceq 1 \preceq (q-1)$, compute $\mathscr{J} = \mathscr{W} \cdot \mathscr{D}$, compute $\mathscr{Z}_c = \mathscr{W}_p \cdot \mathscr{T}_c$, compute $\hbar_2 = H_2(\mathscr{J}, \mathscr{Z}_p, \mathscr{G}\mathscr{D}, \text{ID}_c, \text{PK}_p)$, compute $\hbar_3 = H_3(\mathscr{J}, \mathscr{Z}_p, \mathscr{G}\mathscr{D}, \text{ID}_c, \text{PK}_c)$, compute $\mathscr{B}_p = \mathscr{X}_{p+} \mathscr{W}_p \cdot \hbar_2 + \mathscr{S}_p \cdot \hbar_3 \bmod q$, and compute $\hbar_4 = H_4(\mathscr{Z}_p, \mathscr{J}, \mathscr{T}_c, \text{ID}_c)$. Finally, produce a signcrypted text as $\delta = (\mathscr{J}, \mathscr{B}_p, \hbar_4)$ and send it to the consumer.

7.3. Unsigncryption. When the consumer receives the signcrypted content, it verifies the signature and decrypts the

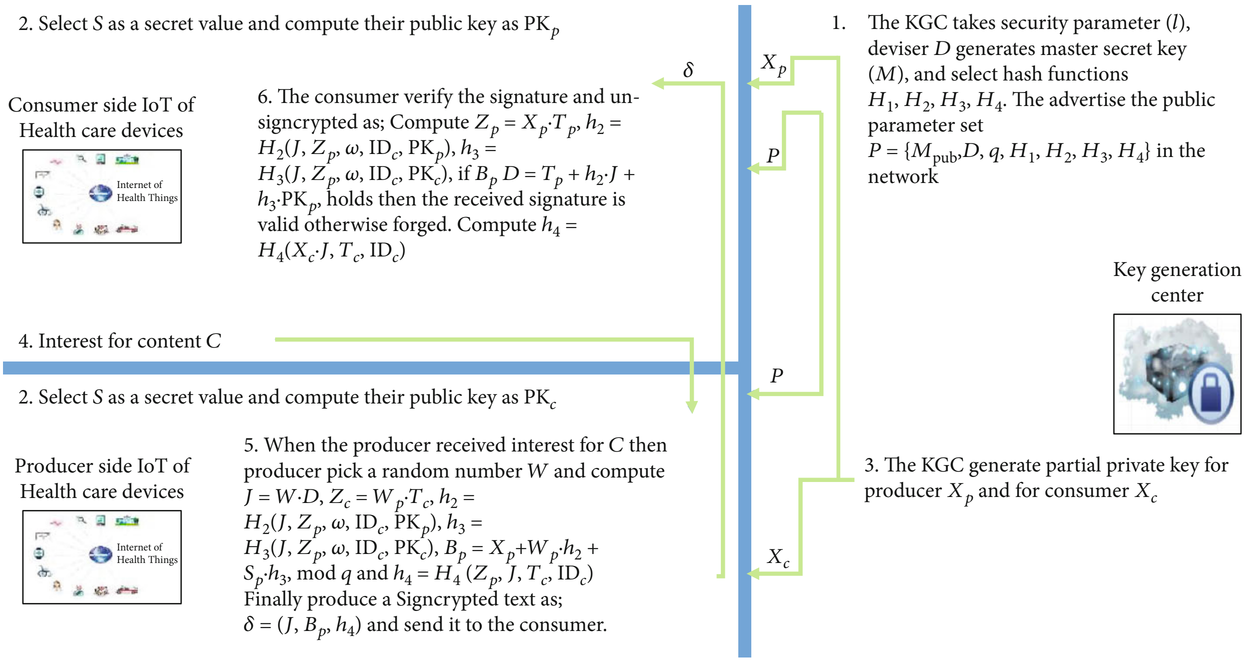

2. Select S as a secret value and compute their public key as PK_p
Consumer side IoT of Health care devices
Internet of Health Things
6. The consumer verify the signature and un-signcrypted as; Compute $Z_p = X_p \cdot T_p$, $h_2 = H_2(J, Z_p, \omega, \mathrm{ID}_c, \mathrm{PK}_p)$, $h_3 = H_3(J, Z_p, \omega, \mathrm{ID}_c, \mathrm{PK}_c)$, if $B_p\, D = T_p + h_2 \cdot J + h_3 \cdot \mathrm{PK}_p$, holds then the received signature is valid otherwise forged. Compute $h_4 = H_4(X_c \cdot J, T_c, \mathrm{ID}_c)$
δ
X_p
P
1. The KGC takes security parameter (l), deviser D generates master secret key (M), and select hash functions H_1, H_2, H_3, H_4. The advertise the public parameter set $P = \{M_{\mathrm{pub}}, D, q, H_1, H_2, H_3, H_4\}$ in the network
Key generation center
4. Interest for content C
P
2. Select S as a secret value and compute their public key as PK_c
Producer side IoT of Health care devices
Internet of Health Things
5. When the producer received interest for C then producer pick a random number W and compute $J = W \cdot D$, $Z_c = W_p \cdot T_c$, $h_2 = H_2(J, Z_p, \omega, \mathrm{ID}_c, \mathrm{PK}_p)$, $h_3 = H_3(J, Z_p, \omega, \mathrm{ID}_c, \mathrm{PK}_c)$, $B_p = X_p + W_p \cdot h_2 + S_p \cdot h_3$, mod q and $h_4 = H_4\,(Z_p, J, T_c, \mathrm{ID}_c)$ Finally produce a Signcrypted text as; $\delta = (J, B_p, h_4)$ and send it to the consumer.
3. The KGC generate partial private key for producer X_p and for consumer X_c
X_c

Figure 6: Deployment of the designed scheme.

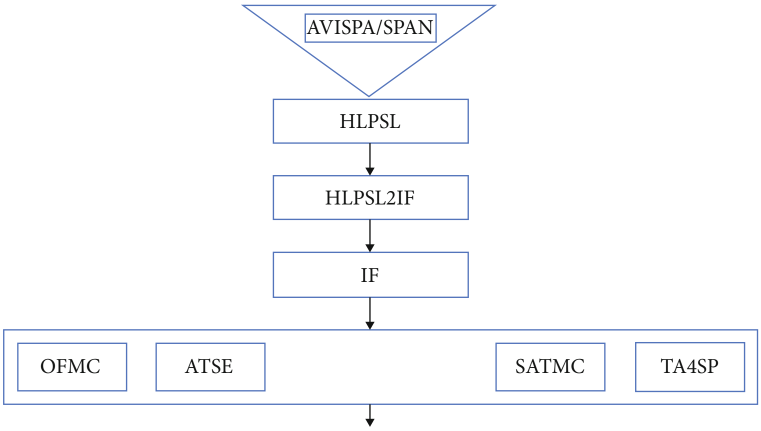


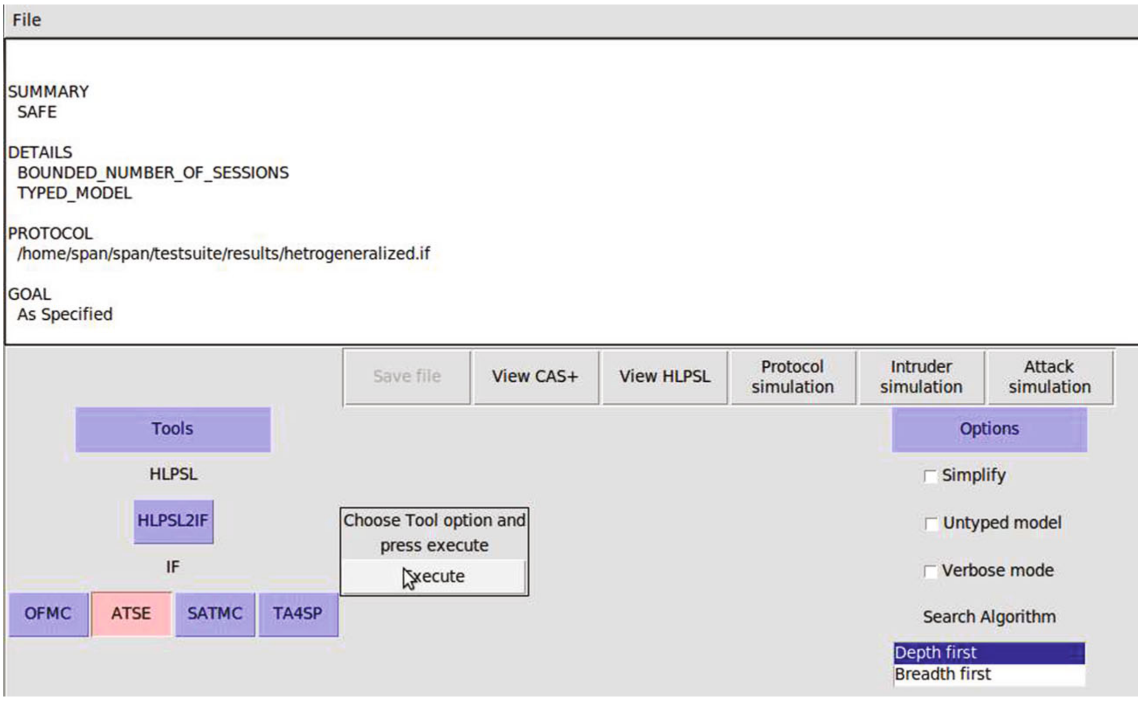
Figure 7: Pushdown flow of AVISPA.

content as compute $\mathscr{Z}_p = \mathscr{X}_p.\cdot\mathscr{T}_p$, compute $\hbar_2 = H_2(\mathscr{J}, \mathscr{Z}_p, \mathscr{W}, \mathrm{ID}_c, \mathrm{PK}_p)$, and compute $\hbar_3 = H_3(\mathscr{J}, \mathscr{Z}_p, \mathscr{W}, \mathrm{ID}_c, \mathrm{PK}_c)$. If $\mathscr{B}_p \mathscr{D} = \mathscr{T}_p + \hbar_2 \cdot \mathscr{J} + \hbar_3 \cdot \mathrm{PK}_p$ holds, then the received signature is valid otherwise forged. Also, the consumer can decrypt $\hbar_4 = H_4(\mathscr{X}_c \cdot \mathscr{J}, \mathscr{T}_c, \mathrm{ID}_c)$.

8. Simulation of the Designed Scheme through AVISPA

AVISPA tool [40] is a top-down automated validation of the security protocols used to verify the resistivity of a given security protocol against replay attacks and man-in-the-middle attacks. AVISPA uses the rule-oriented high-level protocol specification language (HLPSL) [41], to verify the security protocol. The code of the HLPSL is transformed to an intermediate format (IF) via a translator known as HLPSL-2-IF [42]. The IF is then given to the required four backend checkers, namely, OFMC, CL-AtSe, SATMC, and TA4SP. For more details regarding NDN, we recommend readers to study [40]. A generic structure of AVISPA is illustrated in Figure 7.

This section implemented mandatory roles for the session, goals, and environment. We evaluate the newly designed scheme using the two backend checkers of AVISPA such as constraint-logic-based attack searcher (CL-AtSe) and on-the-fly model checker (OFMC) with the help of the graphical user interface (GUI) of security protocol animator (SPAN) [43]. Moreover, for evaluation, AVISPA implements the Dolev-Yao threat model [35]. The simulation results reported in Figures 8 and 9 show the formal verification and security of the designed scheme against man-in-the-middle attacks and replay attacks.



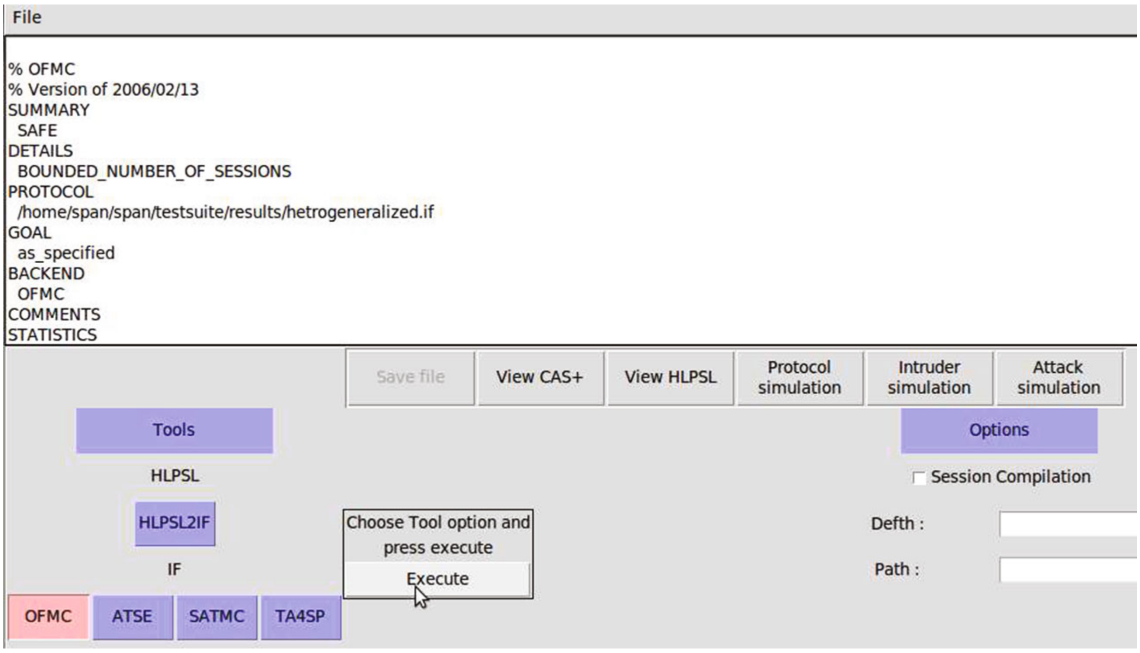
```
SUMMARY
 SAFE

DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
 TYPED_MODEL

PROTOCOL
 /home/span/span/testsuite/results/hetrogeneralized.if

GOAL
 As Specified
```

Figure 8: Proposed scheme simulation results of ATSE.



```
% OFMC
% Version of 2006/02/13
SUMMARY
 SAFE
DETAILS
 BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
 /home/span/span/testsuite/results/hetrogeneralized.if
GOAL
 as_specified
BACKEND
 OFMC
COMMENTS
STATISTICS
```

Figure 9: Proposed scheme simulation results of OFMC.

9. Conclusion

As the number of biomedical devices coupled with the Internet grows, providing strong security with privacy is becoming a prime concern. The overuse of IoHT devices raises a serious issue in the medical domain. Due to the critique and sensitivity of the data within the healthcare domain, proper security and privacy in IoHT undermines patient privacy and endangers patients' lives. However, IoHT transfers data via a public channel, which has implications for security and privacy. Researchers suggest named data networking (NDN), a future Internet model that suits the caregivers and mobile patients to address this issue. Hence, in this article, we have proposed a lightweight certificateless signcryption scheme for NDN-based IoHT. For most IoHT applications, traditional cryptographic algorithms are not practical due to low power-embedded devices' power constraints. For this cause, we use hyperelliptic curve cryptosystem (HCC) which utilizes minimal key size. In addition, after comparing with the relevant schemes, the designed scheme has proven to be effective in terms of cost complexities. For more evidence, we validate the designed scheme attacks' security using the formal verification tool AVISPA.

An extension of the designed scheme is essential that provides simultaneous encryption and signature. We also aim to improve the security of the given scheme by adding some other elements of official formal analysis, such as random oracle model. All these factors are under development stages and will be taken into consideration in the near future.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgments

The authors are grateful to the Taif University Researchers Supporting Project (number TURSP-2020/36), Taif University, Taif, Saudi Arabia. This research work was also partially supported by the Faculty of Computer Science and Information Technology, University of Malaya, under Postgraduate Research Grant PG035-2016A.

References

[1] F. Alsubaei, A. Abuhussein, and S. Shiva, "Security and privacy in the Internet of Medical Things: taxonomy and risk assessment," in *2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops)*, pp. 112–120, Singapore, 2017.

[2] P. Waurzyniak, *Securing Manufacturing Data in the Cloud*, Advanced Manufacturing, 2016.

[3] S. Prakash, "An overview of healthcare perspective based security issues in wireless sensor networks," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, 2016.

[4] G. Hatzivasilis, O. Soultatos, S. Ioannidis, C. Verikoukis, G. Demetriou, and C. Tsatsoulis, "Review of security and privacy for the Internet of Medical Things (IoMT)," in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pp. 457–464, Santorini, Greece, 2019.

[5] A. Afanasyev, J. Burke, T. Refaei, L. Wang, B. Zhang, and L. Zhang, "A brief introduction to named data networking," in *MILCOM 2018-2018 IEEE Military Communications Conference (MILCOM)*, pp. 1–6, Los Angeles, CA, USA, 2018.

[6] R. K. Thelagathoti, S. Mastorakis, A. Shah, H. Bedi, and S. Shannigrahi, "Named data networking for content delivery network workflows," 2020, https://arxiv.org/abs/2010.12997.

[7] L. Zhang, A. Afanasyev, J. Burke et al., "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[8] A. Afanasyev, J. Shi, B. Zhang et al., *NFD Developer's Guide*, Department of Computer Science, University of California, Los Angeles, Los Angeles, CA, USA, 2014.

[9] U. Ali, *RFID Authentication Scheme Based on Hyperelliptic Curve Signcryption*, IEEE Access, 2021, http://ieeexplore-ieee-org.ezproxy.um.edu.my/document/9389538.

[10] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 26, no. 1, pp. 96–99, 1983.

[11] W. P. Wardlaw, "The RSA public key cryptosystem," in *Coding Theory and Cryptography*, pp. 101–123, Springer, 2000.

[12] M. Agren, *On Some Symmetric Lightweight Cryptographic Designs*, Department of Electrical and Information Technology, Faculty of Engineering, 2012.

[13] H. Delfs, H. Knebl, and H. Knebl, *Introduction to Cryptography*, vol. 2, Springer, Heidelberg, 2002.

[14] R. Dutta, R. Barua, and P. Sarkar, *Pairing-Based Cryptography: A Survey*, 2004.

[15] A. Menezes, "An introduction to pairing-based cryptography," *Recent Trends in Cryptography*, vol. 477, pp. 47–65, 2009.

[16] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, pp. 203–209, 1987.

[17] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 2018.

[18] S. Hussain, I. Ullah, H. Khattak et al., "A lightweight and formally secure certificate based Signcryption with proxy re-encryption (CBSRE) for Internet of Things enabled smart grid," *IEEE Access*, vol. 8, pp. 93230–93248, 2020.

[19] S. Hussain, I. Ullah, H. Khattak, M. A. Khan, C. M. Chen, and S. Kumari, "A lightweight and provable secure identity-based generalized proxy signcryption (IBGPS) scheme for Industrial Internet of Things (IIoT)," *Journal of Information Security and Applications*, vol. 58, p. 102625, 2021.

[20] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption) cost (signature)+cost (encryption)," in *Annual International Cryptology Conference*, pp. 165–179, Berlin, Heidelberg, 1997.

[21] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Workshop on the Theory and Application of Cryptographic Techniques*, vol. 196, pp. 47–53, Heidelberg, 1984.

[22] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *International conference on the theory and application of cryptology and information security*, vol. 2894, pp. 452–473, Berlin, Heidelberg, 2003.

[23] M. Barbosa and P. Farshim, "Certificateless signcryption," in *2008 ACM Symposium on Information, Computer and Communications Security*, pp. 369–372, ACM, New York, 2008.

[24] A. Mehmood, I. Noor-Ul-Amin, and A. I. Umar, "Public verifiable generalized authenticated encryption based on hyper elliptic curve," *Journal of Applied Environmental and Biological Sciences*, vol. 7, pp. 194–200, 2017.

[25] C. Zhou, G. Gao, and Z. Cui, "Certificateless signcryption in the standard model," *Wireless Personal Communications*, vol. 92, pp. 495–513, 2016.

[26] P. Rastegari and M. Berenjkoub, "An efficient *certificateless signcryption* scheme in the standard model," *ISeCure*, vol. 9, pp. 3–16, 2017.

[27] H. Yu and B. Yang, "Pairing-free and secure certificateless signcryption scheme," *The Computer Journal*, vol. 60, no. 8, pp. 1187–1196, 2017.

[28] C. Zhou, "Certificateless signcryption scheme without random oracles," *Chinese Journal of Electronics*, vol. 27, no. 5, pp. 1002–1008, 2018.

[29] W. Luo and W. Ma, "Secure and efficient data sharing scheme based on certificateless hybrid signcryption for cloud storage," *Electronics*, vol. 8, p. 590, 2019.

[30] X. Liu, Z. Wang, Y. Ye, and F. Li, "An efficient and practical certificateless signcryption scheme for wireless body area networks," *Computer Communications*, vol. 162, pp. 169–178, 2020.

[31] P. Kasyoka, M. Kimwele, and S. M. Angolo, *Cryptoanalysis of a Pairing-Free Certificateless Signcryption Scheme*, ICT Express, 2020.

[32] F. Li, J. Hong, and A. A. Omala, "Efficient certificateless access control for industrial Internet of Things," *Future Generation Computer Systems*, vol. 76, pp. 285–292, 2017.

[33] C. Zhou, Z. Zhao, W. Zhou, and Y. Mei, "Certificateless key-insulated generalized signcryption scheme without bilinear pairings," *Security and Communication Networks*, vol. 2017, Article ID 8405879, 17 pages, 2017.

[34] G. Swapna, K. A. Ajmath, and G. Thumbur, "An efficient pairing-free certificateless signcryption scheme with public verifiability," *Journal of Mathematics and Computer Science*, vol. 11, no. 1, pp. 24–43, 2020.

[35] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[36] S. Hussain, S. S. Ullah, A. Gumaei, M. Al-Rakhami, I. Ahmad, and S. M. Arif, "A novel efficient certificateless signature scheme for the prevention of content poisoning attack in named data networking based Internet of Things," *IEEE Access*, vol. 9, pp. 40198–40215, 2021.

[37] D. Saxena, V. Raychoudhury, and N. Sri Mahathi, "SmartHealth-NDNoT: Named Data Network of Things for healthcare services," *MobileHealth@ MobiHoc*, pp. 45–50, 2015.

[38] D. Saxena and V. Raychoudhury, "Design and verification of an NDN-based safety-critical application: a case study with smart healthcare," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 5, pp. 991–1005, 2017.

[39] X. Wang and S. Cai, "Secure healthcare monitoring framework integrating NDN-based IoT with edge cloud," *Future Generation Computer Systems*, vol. 112, pp. 320–329, 2020.

[40] L. Vigano, "Automated security protocol analysis with the AVISPA tool," *Electronic Notes in Theoretical Computer Science*, vol. 155, pp. 61–86, 2006.

[41] D. von Oheimb, "The high-level protocol specification language HLPSL developed in the EU project AVISPA," in *Proceedings of 3rd APPSEM II (Applied Semantics II) Workshop (APPSEM'05)*, pp. 1–17, Germany, 2005.

[42] S. S. Ullah, I. Ullah, H. Khattak et al., "A lightweight identity-based signature scheme for mitigation of content poisoning attack in named data networking with Internet of Things," *IEEE Access*, vol. 8, pp. 98910–98928, 2020.

[43] S. S. Ullah, S. Hussain, A. Gumaei, and H. AlSalman, "A secure NDN framework for Internet of Things enabled healthcare," *Computers, Materials & Continua*, vol. 67, no. 1, pp. 223–240, 2021.

Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5584667, 14 pages
https://doi.org/10.1155/2021/5584667

Research Article

An IoT-Based Network for Smart Urbanization

Sabeeh Ahmad Saeed,[1] Farrukh Zeeshan Khan,[2] Zeshan Iqbal,[2] Roobaea Alroobaea,[3] Muneer Ahmad,[4] Muhammad Talha,[5] Muhammad Ahsan Raza,[6] and Ihsan Ali[4]

[1]*Department of Computer Science, COMSATS University Islamabad, WAH Campus, Pakistan*
[2]*Department of Computer Science, University of Engineering and Technology, Taxila, Pakistan*
[3]*Department of Computer Science, College of Computers and Information Technology, Taif University, P. O. Box 11099, Taif 21944, Saudi Arabia*
[4]*Faculty of Computer Science & Information Technology, Universiti Malaya, 50603 Kuala Lumpur, Malaysia*
[5]*Deanship of Scientific Research, King Saud University, Riyadh, Saudi Arabia*
[6]*Department of Information Technology, Bahauddin Zakariya University Multan, Pakistan*

Correspondence should be addressed to Ihsan Ali; ihsanalichd@siswa.um.edu.my

Received 4 February 2021; Revised 4 March 2021; Accepted 18 March 2021; Published 5 April 2021

Academic Editor: Habib Ullah Khan

Internet of Things (IoT) is considered one of the world's ruling technologies. Billions of IoT devices connected together through IoT forming smart cities. As the concept grows, it is very challenging to design an infrastructure that is capable of handling large number of devices and process data effectively in a smart city paradigm. This paper proposed a structure for smart cities. It is implemented using a lightweight easy to implement network design and a simpler data format for information exchange that is suitable for developing countries like Pakistan. Using MQTT as network protocol, different sensor nodes were deployed for collecting data from the environment. Environmental factors like temperature, moisture, humidity, and percentage of CO_2 and methane gas were recorded and transferred to sink node for information sharing over the IoT cloud using an MQTT broker that can be accessed any time using Mosquitto client. The experiment results provide the performance analysis of the proposed network at different QoS levels for the MQTT protocol for IoT-based smart cities. JSON structure is used to formulate the communication data structure for the proposed system.

1. Introduction

Internet of Things (IoT) is considered to be another important Information Communication Technology (ICT) wave after the invention of personal computers (PCs), Ethernet, Internet, and the cellular communication [1]. IoT has taken over the world since 2005 and became the very core of the future economic developments in the field of Internet, communication, and networking [2]. Many countries around the world have taken into account IoT as part of national strategy for sustainable development in their governmental and general public sectors by completing the logical and conceptual studies at service level. For example, Japan's broadband access is based on ubiquitous and people-oriented technology, providing services with an objective to help efficient communication between people and people and things and between things as well [3]. The South Korean smart home automation systems help the residence to control many home appliances remotely and also enjoy bidirectional multimedia services [4]. Singapore is also second to none; her next-generation I-hub main objective is to provide secure next-generation "U"-type networks through ubiquitous networking [5]. All these and other such similar projects have laid the foundation of IoT firmly around the world.

IoT domains include healthcare, industry, transportation, education, and emergency response to any sort of natural or man-made disasters under stressed conditions. IoT enables the people to interact (see, hear, and think) with the sensors that help them to share information, make intelligent decisions, and respond to queries efficiently. In other words, IoT helps to see, hear, and think an environment from an eye of technology (sensors). On the other hand, IoT also changes traditional devices into smart objects by changing the underlying technologies (ubiquitous, embedded systems, sensor networks, communication technologies, internet protocols, applications, and pervasive computing) of these systems. To better understand the concept of IoT, it is important to learn about the elements of IoT. The related examples and categories of each element are listed and discussed in [6–22]. The major elements of an IoT network are identification, sensing, communication, composition, services, and semantics.

The IoT's main objective is to make Internet, communication, and networking more interesting, impressive, and persuasive by providing easy interaction with a wide variety of devices. This paradigm is spread along a vast set of fields and covers almost all human domains of profession [23]. These complex scenarios develop the particular interest of smart urbanization in an IoT paradigm. Thus, the concept of "smart cities" is born [24]. Smart urbanization is capable of optimizing traditional public service processes by increasing the percentage benefit gained from many services like transportation, lightening, maintenance, surveillance of public areas, preservation of history and cultural heritage, parking, automation of industries, education, hospitals, garbage collection, and many more [1, 25]. According to Pike Research [23] conducted on smart cities, it was reported that by 2020, the estimated market share of smart city will be more than hundred billion dollars with an annual expenditure of just 16 billion. Smart city industry emerges from the interconnection between key industries and the service sectors and form several smart city sectors like smart governance, smart utilities, smart buildings, smart environment, and smart mobility [26]. There are a number of reasons that hinder the growth of smart cities to full capacity, including political, financial, and technical barriers. Authors in [27–30] have discussed several solutions related to these problems. The contribution of this research is to design a system for developing smart cities in underdeveloped countries using IoT technology. The system is easy to use and easy to implement as it uses simple technology, lightweight communication mechanism, and cheap technology. The proposed technology is a suitable small business organization with less budget to spend on their technological needs. Moreover, it is also feasible for governments having low budget to spend on technology adaption.

The rest of the paper is structured as follows: Section 2 provides the review of the already developed technologies; Section 3 provides the detail of the proposed system; Section 4 provides the detail on the working of the system; Section 5 provides the deployment of the system, data collection mechanism simulation implementation, and simulation results; and Section 6 concluded the overall research and provide future directions.

2. Literature Review

In this section, different smart cities around the world are discussed briefly. These projects also provide foundations for the realization of smart cities in underdeveloped countries.

The SyMPHOnY project [31] was a smart city project designed using SIP protocol. A special hardware called the MTCG node was designed as a communication device using the SIP protocol for data transmission. The MTCG node receives data from many sensory devices, like home appliances, temperature sensors, and humidity sensors, using a wireless M-BUS interface and transmits it to a SIP server. The data can be accessed using a SIP client [31–33]. The MTCG node was a set of different packages joined together in a single package device called a core. Eco-U-CITY [34] was the first South Korean project for smart city implementation. The project was completed in 2008 to convert the cities of Hwaseong and Dengtan into smart cites. Eco-U-CITY is a project based on green technology, for better safety and comfort of the public. The major aim of this project was to use green technology to reduce the emission of carbon contents in the atmosphere. The project was implemented using a specially developed system called "An Integrated Service Management Platform" (ISMP). ISMP is a 3-layered model for smart cities presented in [35]. The layers present in an ISMP system are service layer, middle-ware layer, and the infrastructure layer.

Under the supervision of the New York City Mayor's Office, the city launched the New York Digital City Program sponsored by the Mayor's Office itself. The program was based on an IT-driven portal called the NYC.GOV portal, with an aim of combining all city's general public on a single platform, i.e., the portal. All citizens were able to access all the services, functions, and applications through their smart devices, mobile phones, and commercial social media. Barcelona Smart City Program implements a three-layer model for a wide variety of technical capabilities stated in [36]. The first layer constitutes sensors, the second layer of the model was based on a City Operating System (City OS), and the third layer was comprised of customer interface. It was using ICT throughput throughout the implementation and development of smart cities using smart city models presented in [37–39]. The city has started a series of projects supporting the concept of smart city, over a physical network covering more than 500 kilometers of area via fiber optics. The city project is aimed at integrating telecommunication and Internet together using twelve initiatives identified via used smart city models. The project has four stages and has successfully completed its three stages, and the fourth stage is under process [39]. Padova Smart City [26] was an implementation of urban IoT concept in the Padova city with the alliance of public and private bodies of the city. The major parties of the alliance were the municipality of Padova, the Department of Information Engineering University of Padova, and Patavina Technologies. The first provides the financial assistance required to aid the project; the second provides the background for the project to start and also give its feasibility report. Finally, the third party, which is a spin-off of the University of Padova and is specialized in the development of creative IoT solutions, provided design and implementations for the IoT nodes and the software required to control the network.

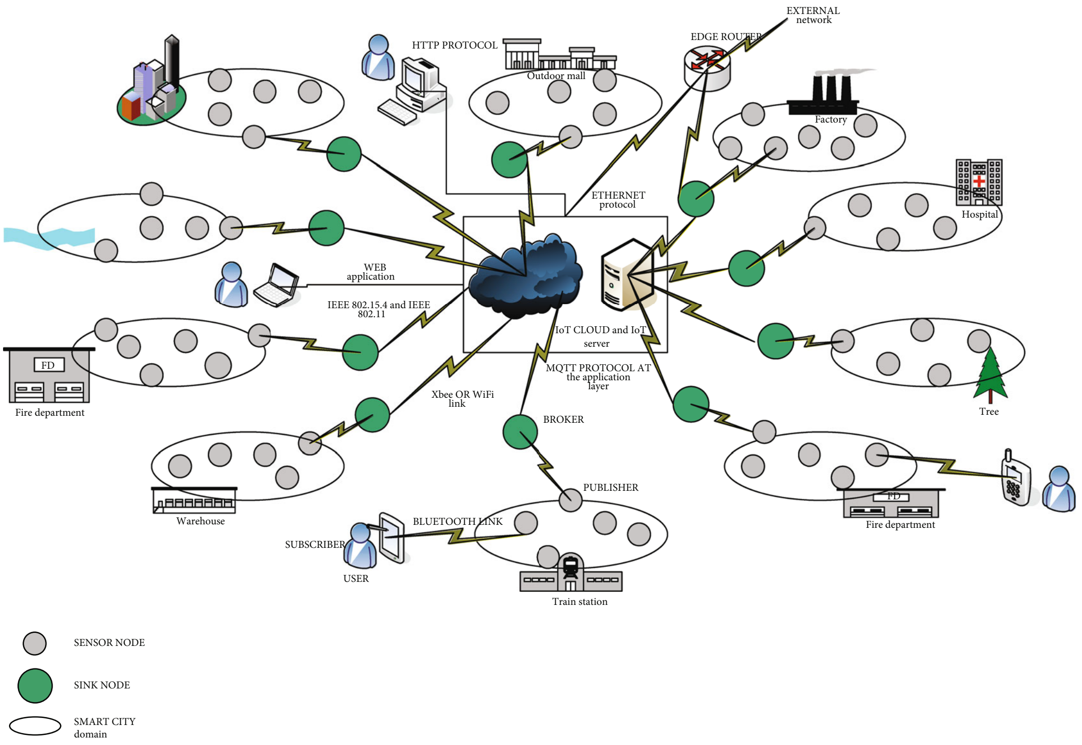


Figure 1: Proposed network architecture.

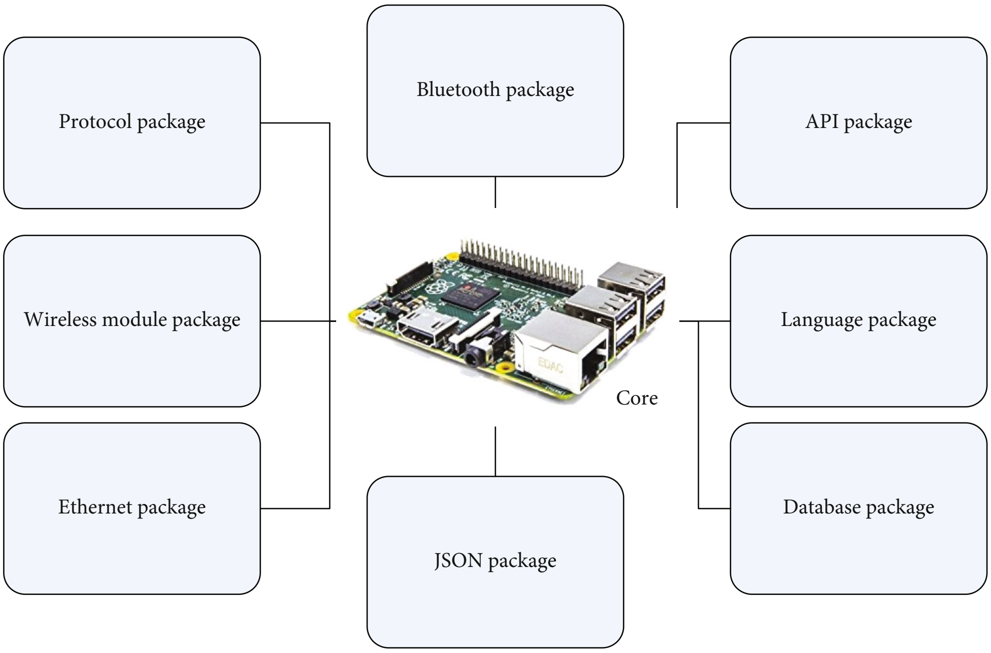


Figure 2: Proposed framework of the sink node.

3. Proposed IoT Network Design

Keeping in mind the study done in the literature review and different information provided in different research papers and research projects, the proposed network architecture used to carry out our research is given Figure 1. It is evident from Figure 1 that the proposed network architecture consists of the following network entities.

The proposed network consists of several devices connected together. The devices are classified into sensor nodes, sink nodes, edge router, IoT cloud, and end user all discussed in the later paragraphs. The network is formed when several

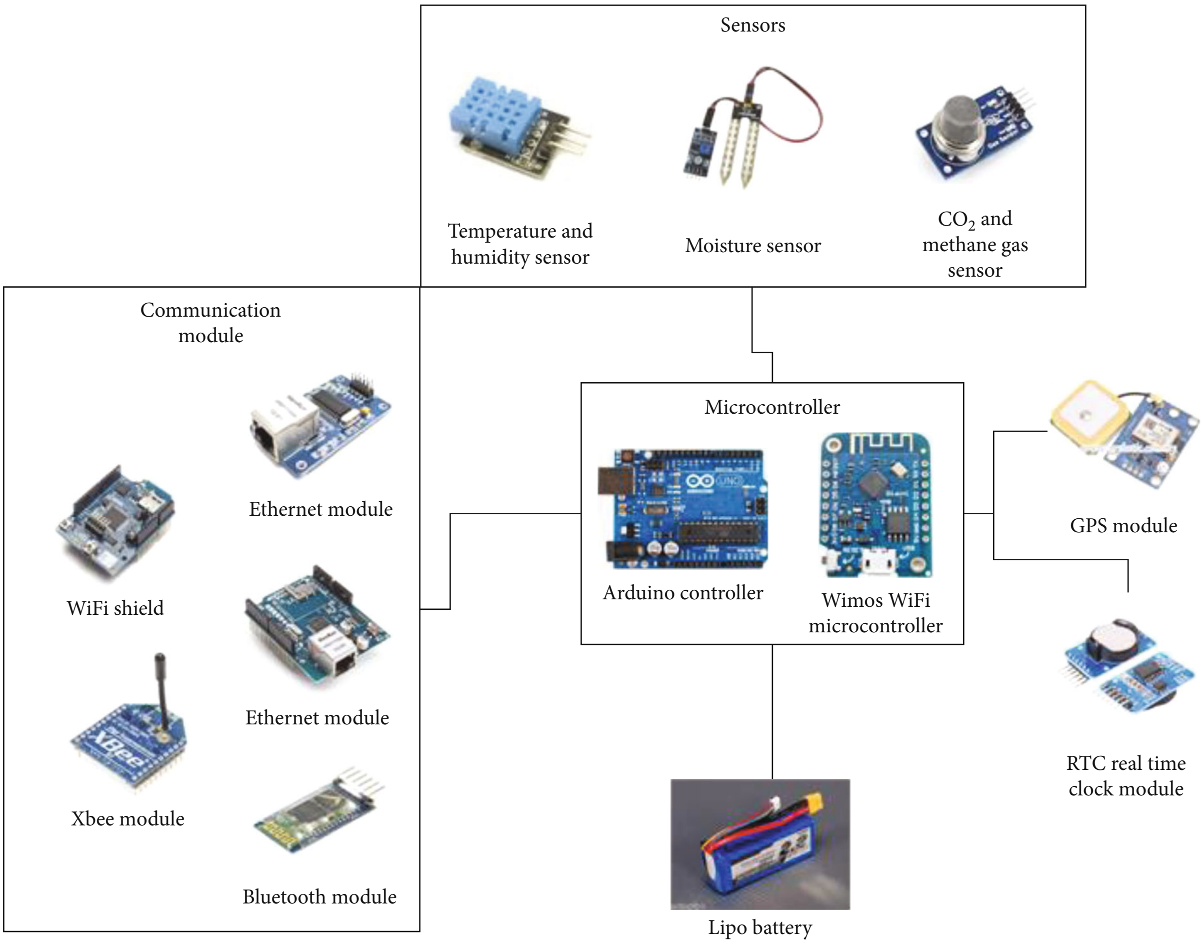


Figure 3: Block diagram of sensor node with main components.

```
Algorithm:
  Structure Reading {
          "Sensor" :{
              "System":
                  "Type": {
                      "Name": "Sensor Name",
                      "ID": "value"
                  },
                  "Placement of the Sensor": {
                      "Latitude": "value",
                      "Longitude": "value",
                  },
      },
          "Read_Value": ["Reading Type": "Reading Name", "Value": "value", "units":
      "value"],
      "Time": "Time Stamp",
      "Date": "Date Stamp",
      "Status": {
          "Name": "Device Status",
          "Value": {"NULL =0", "OK =1", "ERROR =2", "UNKNOWN =3"},
    };
}
```

Algorithm 1: Proposed data structure format.

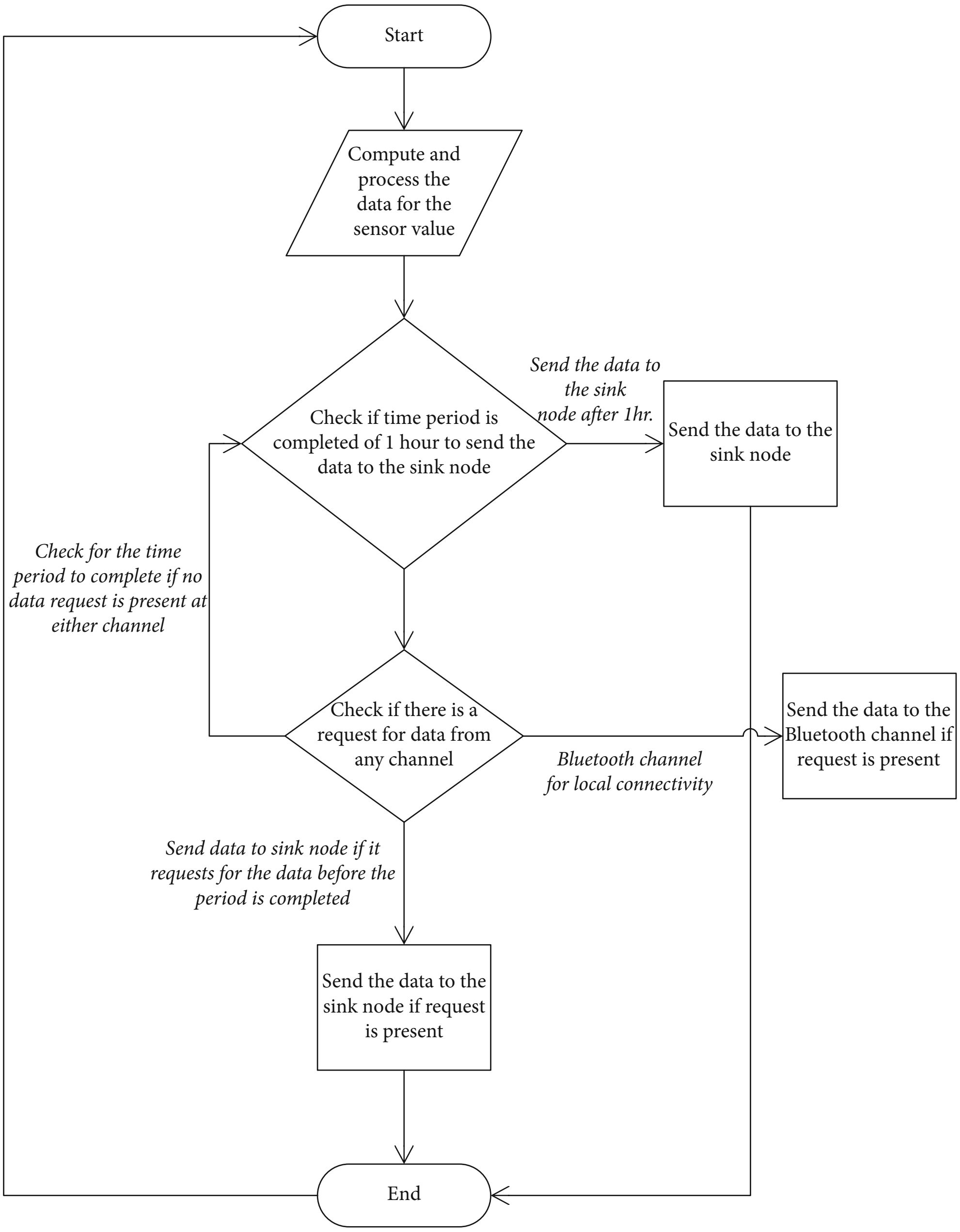

Start
Compute and process the data for the sensor value
Check if time period is completed of 1 hour to send the data to the sink node
Send the data to the sink node after 1hr.
Send the data to the sink node
Check for the time period to complete if no data request is present at either channel
Check if there is a request for data from any channel
Bluetooth channel for local connectivity
Send the data to the Bluetooth channel if request is present
Send data to sink node if it requests for the data before the period is completed
Send the data to the sink node if request is present
End

Figure 4: Sensor node activity.

sensors deployed connect to a common sink node that acts as a broker to collect and share information. Sensor nodes continuously observe the deployed vicinity for recording data related to several environmental attributes like temperature, humidity, moisture, and gaseous content percentage in the atmosphere; the sensor node then transfers the data to the sink node periodically that keeps data safe for uploading the information to the IoT cloud after a regular interval or whenever a demand for any particular information or reading is received. The IoT cloud provides user an interface to observe and keep track of the changes occurring in any location by keeping a periodic track of the information received from the sink node.

The *sink node* is implemented using a Raspberry Pi board 2 with a Windows 10 IoT core. The core supports the bundles for using Java Programming Environment. Arduino IDE sketch is also installed on the system to support Arduino. The Mosquitto version 1.4.9 which is an open-source MQTT server was deployed on an Amazon Web Server. Specifications of the Amazon Web Server are as follows: AWS service: EC2, instance type: t2. Micro, OS: Windows server 2016 base1 virtual CPU, storage: 30 GB, and RAM: 1 GB. Node-RED is a visual tool for wiring IoT devices. Node-RED provides web interface, which can send commands to the MQTT server. Node-RED provides interaction between clients and server. Node-RED offers a browser-based flow editor to wire together flows by applying a broad range of nodes in the palette. Flows can be then implemented dynamically in a single click. Due to built-in library in node-RED, useful functions and flows can be saved for reuse. The Bluetooth module is

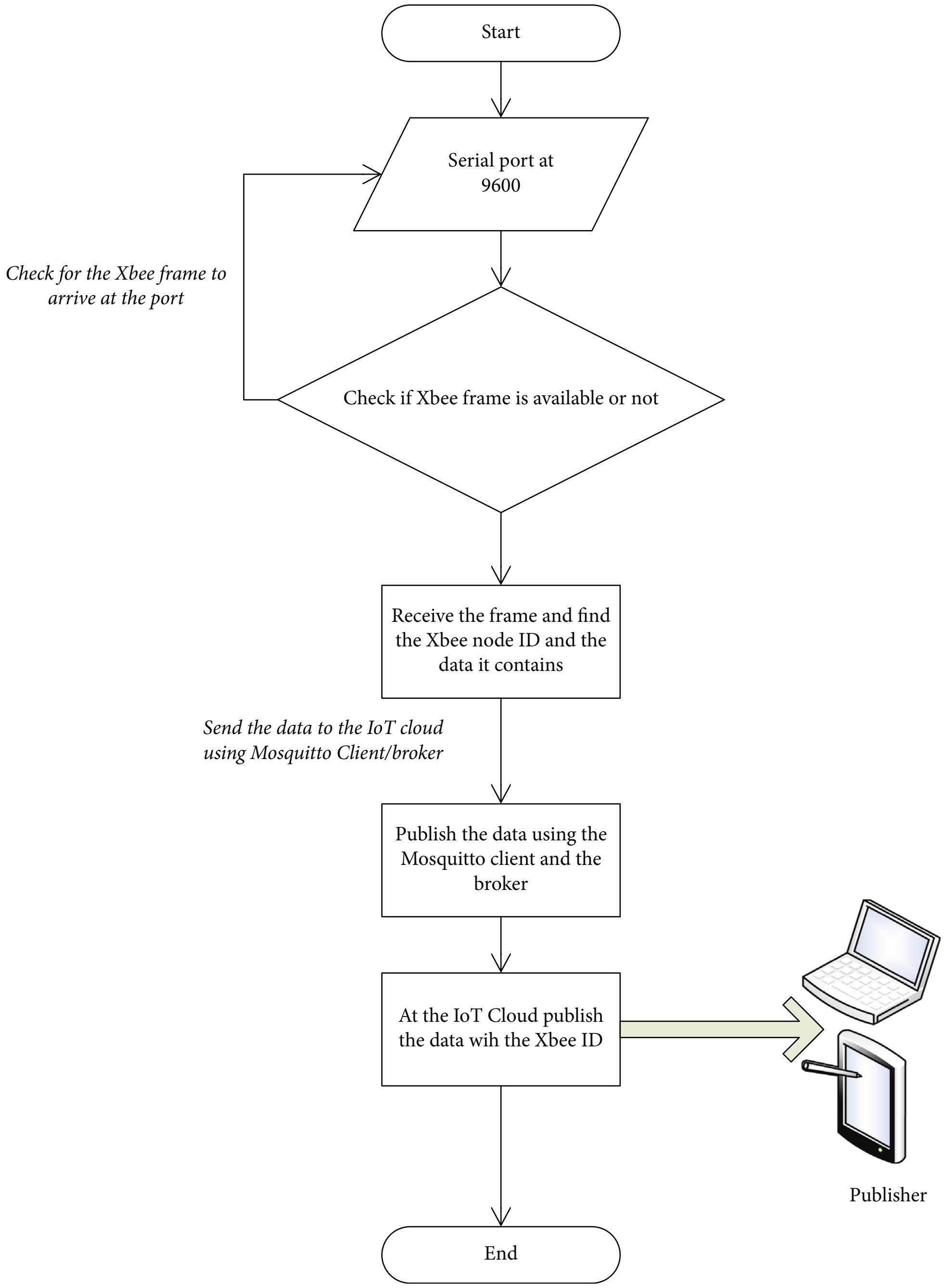

Start
Serial port at 9600
Check for the Xbee frame to arrive at the port
Check if Xbee frame is available or not
Receive the frame and find the Xbee node ID and the data it contains
Send the data to the IoT cloud using Mosquitto Client/broker
Publish the data using the Mosquitto client and the broker
At the IoT Cloud publish the data wih the Xbee ID
Publisher
End

Figure 5: Publisher.

also attached to the sink node to connect any local user to receive any information using a mobile application data can be designed to access data from the Bluetooth module. The framework used for the proposed system is shown in Figure 2.

As shown in Figure 2, the framework consists a core package including a powerful board with some operating or the real-time operating system. The core is attached to several bundles depending on the needs of the system. The major bundles included in the framework are as follows: core bundle: the main package that controls the communication and other bundles present in the framework; protocol bundle: the package containing the definition and implementation of the said protocols for the proposed network; wireless module bundle: the package containing the possible definition of any wireless module used for connecting the sink node to the other devices; Ethernet bundle: the package for installing and using Ethernet in the framework; JSON bundle: the package giving the information regarding the data structure format used for receiving and extracting the required information from the data received from the sensor nodes; database bundle: the package giving information about the used database in the proposed research; language bundle: the package giving information about the programming language used; and

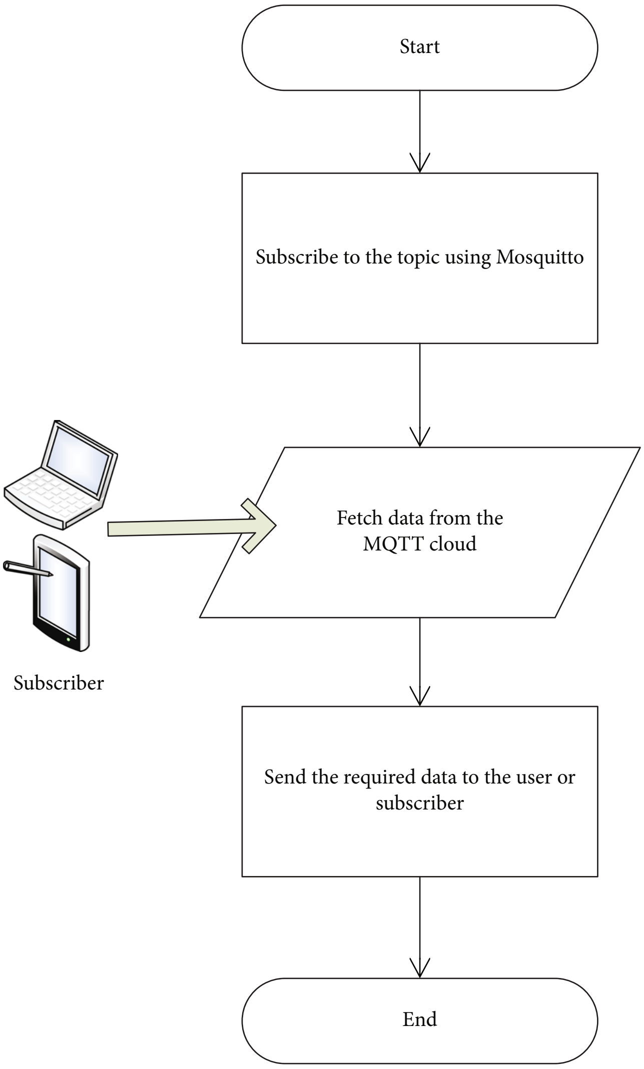

Start
Subscribe to the topic using Mosquitto
Fetch data from the MQTT cloud
Subscriber
Send the required data to the user or subscriber
End

Figure 6: Publisher.

API bundle: the package containing the cloud API's and other API's used to provide communication ease and development ease.

The control system of the *sensor node* is implemented using an Arduino microcontroller (any of UNO, MEGA, Mini, and Micro). The sensors used for measuring different environment values are Arduino compatible temperature sensor DHT-11 temperature and humidity sensor; the sensor used to measure the CO_2 and methane level is the Arduino compatible MQ-135 gas sensor. The sensor used for moisture measurement is the Arduino compatible HL-69 soil moisture sensor. The communication module used is the Xbee module using IEEE 802.15.4 Standard and forms the core of the network. The other module used to form the core network is the WiFi shield for Arduino ESP-8266-WRL-13287. The HC-05 Bluetooth module is used to provide local connectivity with the sensor node. Arduino Ethernet shield Wiznet-W5100 or the ENC-28J60 Ethernet module can also be used to provide wired connectivity that is an optional part of the sensor node. Arduino-based RTC DS-3231 real-time clock and the Arduino-based Global Positioning System (GPS) module NEO-6M is used to provide additional information related to the sensor regarding date, time, and location of the sensor. A 12-volt Lipo battery is attached to the sensor node to provide the power to the complete system. The complete hardware package is installed in a box for safe keeping and to preserve it from sever atmospheric effects. This system can also be achieved using a We MOs D1 Mini ESP8266 a wireless 802.11 (Wi-Fi) microcontroller development board. Its key features are as follows: micro-USB, compatible with Arduino, microprocessor: ESP-8266EXNr, pin:/input/output 11 pin, one input pin, operating voltage is 3.3 V, frequency: 80 MHz/160 MHz, and of 4 Mb flash memory. The overall system is illustrated in Figure 3.

Edge router is a network layer device used to link the proposed network to the underlying external network. The edge router if deployed inside the sink node is called as inner edge router. Or when deployed outside the sink node is called as external edge router. The *IoT cloud or the IoT server* is responsible for joining all the sensor/sink networks at different environments at different areas. These entities provide the central point and are the core of the network architecture. A *user* is a person or a machine or an application that requires the data generated from the sensor nodes for some information or just for record keeping or for making any useful decision using the information generated through the proposed system.

Data structure format used for transmitting the data between the network devices is based on the information provided by the JSON structure as presented in [40–44]. The structure format is consisted of a structure having various variable strings, values, and arrays to represent several values received from the sensor node. The major values that are received from the sensor node include the information regarding the sensor: type of sensor, placement of the sensor, and the sensor ID. The second information that is received from the sensor node includes the reading that is generated at the sensor: numeric value and the unit. The third value received from the sensor node is the time and date stamp, and finally, the fourth reading received from the sensor node is the status of the node. The data structure for each of the values received from the sensor node is given below. The structure format for declaring strings, numeric variables, arrays, structures, and objects is the same as described by the JSON structure. The compiled form of the structure is called as the reading structure that contains four subparts. Each part has its own value depending on the data received from the sensor node. The structure is shown in Algorithm 1.

4. Working of the Proposed System

The working algorithm of the structure is divided into three stages shown in Figure 4, Figure 5, and Figure 6, respectively. The sensor waits for the change in the value that it is reading; upon successful reading, a signal is generated and a value is produced. This value is transmitted (published) to the broker using Xbee communication module. At the broker, the packet is received and the value is checked; if found correct, it is written on the file for record keeping. At the broker, if any user requests to subscribe to the data, the broker writes

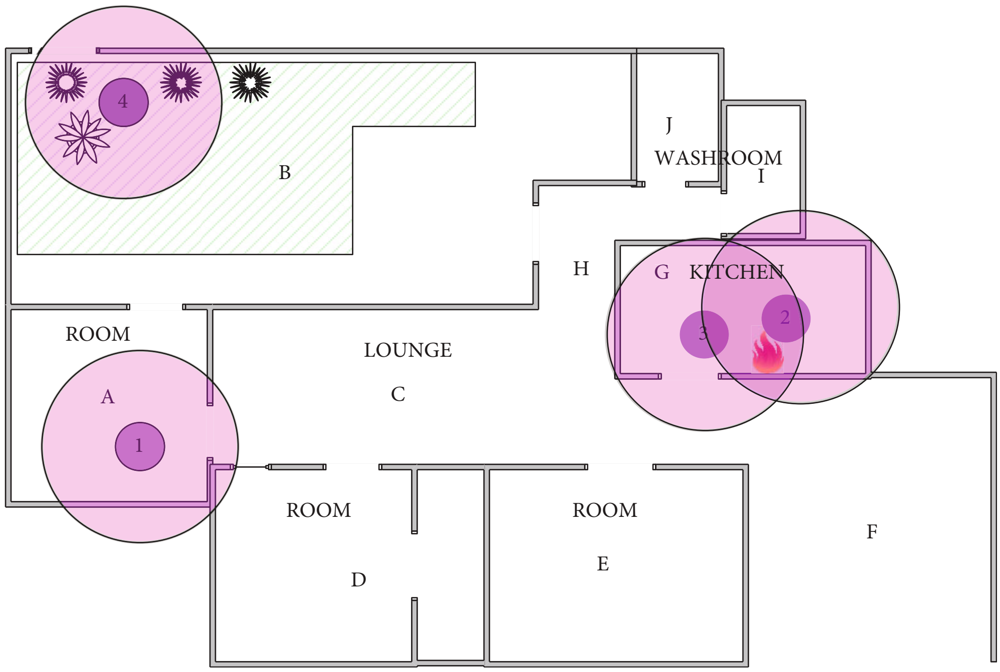


Figure 7: Area of implementation.

Table 1: Simulation parameters.

Protocol/layer	Parameter/option
MQTT	QoS-1, QoS-2, QoS-3
	Payload size
	Publisher's sending rate
	Number of MQTT clients
TCP	Timestamp option
	Maximum segment size (MSS)
MAC layer	IEEE 802.11 MAC
Physical layer	Bandwidth
	Propagation delay
	Error rate
	Error burst

the data to the user who is subscribing to the data. The communication is carried out in three stages that are as follows.

4.1. Stage 1. At the sensor node, the value of the reading under consideration is computed. The sensor node then waits for the time period to expire after 1 hour to send the recorded value to the sink node. If at any time, the sensor node receives a request from the data from the local Bluetooth channel or from the Xbee channel, then the sensor transmits the data to the channel for the request. The Bluetooth channel has priority higher than that of Xbee channel, and if the Xbee channel has a data request, then the time period is restarted. The overall process is shown in Figure 4.

4.2. Stage 2. When data reaches at the sink node, the sink node also waits for an hour before publishing the data at the cloud using the MQTT Mosquitto broker. If there is already a subscription request present at the broker for the data, the sink node then immediately publishes the data at the cloud with the Xbee ID from which data is received at the sink. The process is illustrated in Figure 5.

4.3. Stage 3. Any published data can be subscribed from the IoT cloud using the MQTT Mosquitto client. The process is shown in Figure 6.

The placement of the modules in the experimental area is shown in Figure 7. To publish data at the broker, the sensor nodes are to take several readings from the surrounding and convert them into signals to form a measurable reading. This measurable reading is then is sent to the broker using the Xbee communication module. At the broker, the program waits until an Xbee packet is received or not; when a packet is received, the program extracts the required value from the packet; and using file handling technique, the value along with some additional information is recorded into a file with an extension (.CSV or.TXT). These files are then copied to the MS-Excel sheet, and the graphs are plotted for different sensor values.

4.4. Network Simulation. For the performance evaluation of the proposed network, the proposed network is simulated in OMNET++ and results are analyzed using Wireshark that supports TCP for wireless network models and also supports MQTT protocol. The QoS-0 and QoS-1 described for the MQTT are used to evaluate the end-to-end delay and message delay in the network. To evaluate the performance of the MQTT server, 500 clients were dynamically created. All clients competed for the connection to the server. Once a client gets connected to the server, it sends request to the server and when it receives response from the server, it terminates its connection from the server and hence, a new client

Table 2: Simulation parameters values.

Protocol/layer	Parameter/option
MQTT	QoS-1, QoS-2, QoS-3
	Payload size = 100 bytes
	Publisher's sending rate = 250 Kbits/s
	Number of MQTT clients = 1000
TCP	Timestamp option = yes
	Maximum Segment Size (MSS) = 536
MAC layer	IEEE 802.11 MAC
Physical layer	Bandwidth = 2.4 GHz
	Propagation delay = $T\_air(100) = (25 + 100) * 32\ \mu s = 4.000$ ms
	Error rate = 1%
	Error burst = 0.25

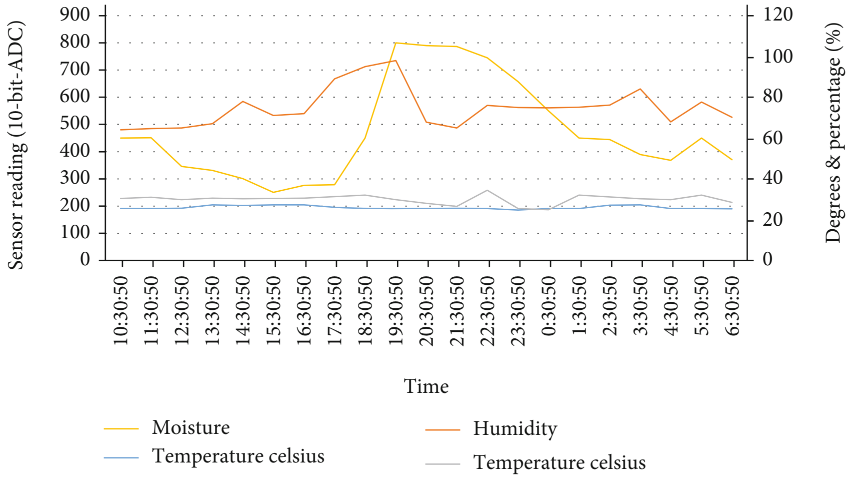


Figure 8: Data collected from sensor nodes.

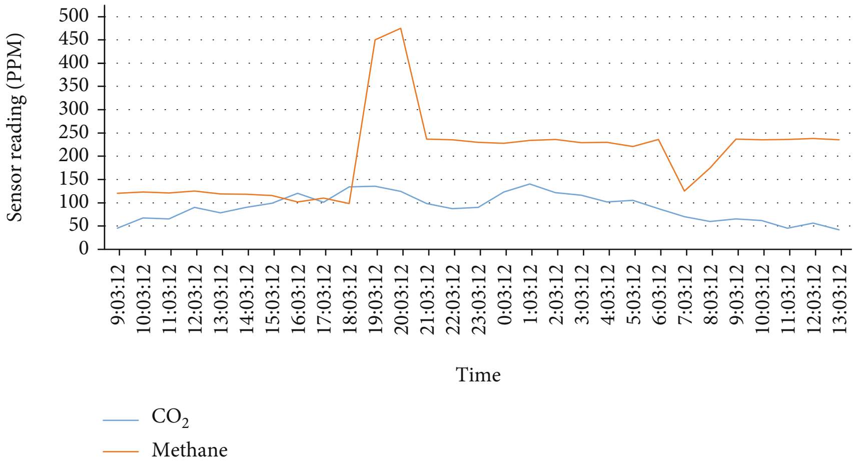


Figure 9: CO_2 and methane gas reading in ppm taken from sensor node 3.

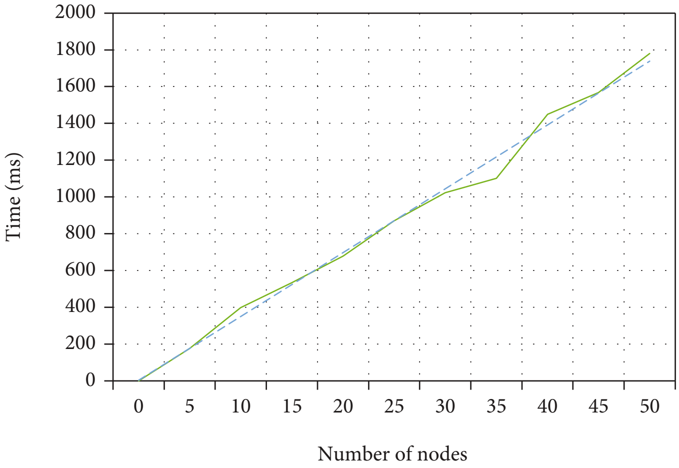


FIGURE 10: Mean connection time delay of network nodes.

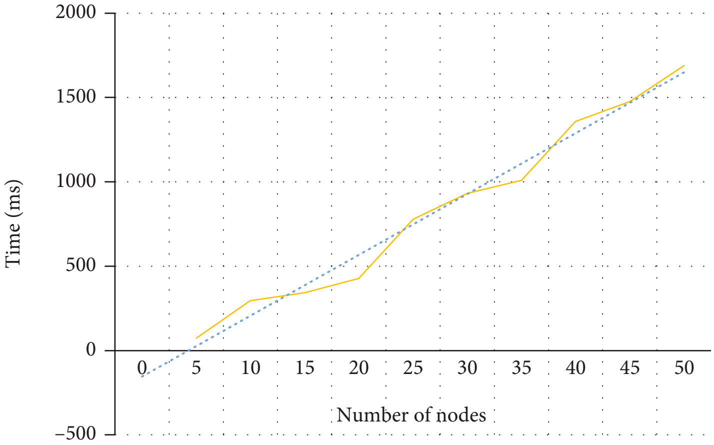


FIGURE 11: Mean response time delay of server.

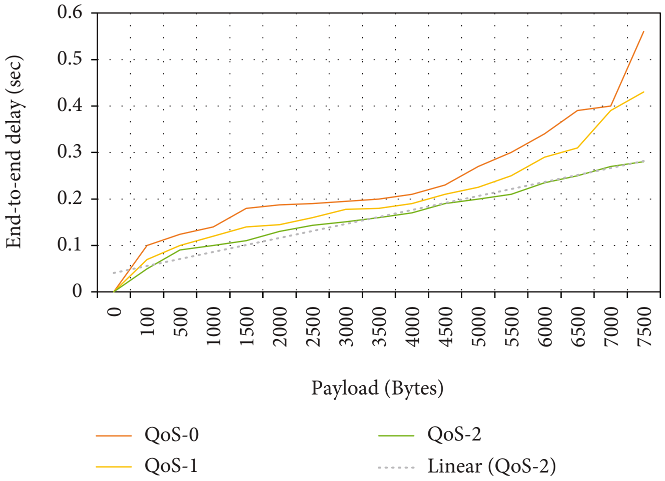


FIGURE 12: Mean end-to-end delay in the network under QoS.

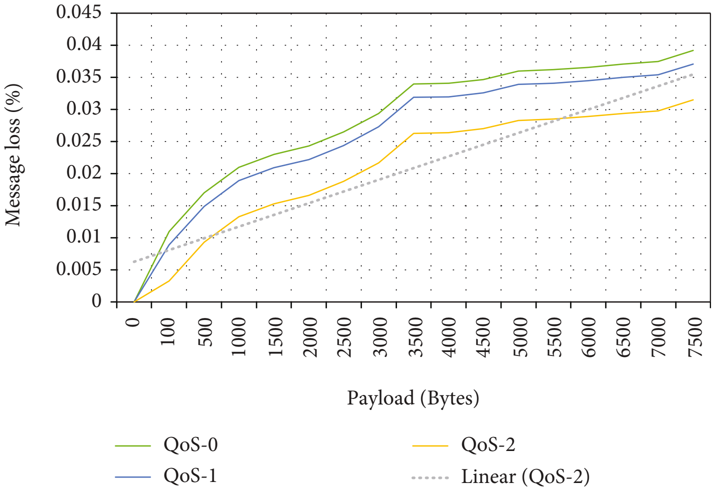


Figure 13: Mean message loss in the network under QoS.

competes for the connection to the server. Here, we observed delay as the difference of time when client sends request and when client receives response from the MQTT server as shown in Equation (1). We also considered delay, experienced in connection time of clients with the server. To get the promising results, experiments were performed in two different scenarios by varying the number of devices and the number of clients generated on these devices and by making changes in their connection times. Simulation parameters used to run the simulation and to calculate results are shown in Table 1.

$$D_{\text{ete}} = \text{RT}_{\text{r}} - \text{ST}_{\text{s}}, \tag{1}$$

where D_{ete} is end-to-end delay of the n^{th} client, RT_{r} time the response was received, and ST_{s} time the request was sent.

The experiment was performed for 1,000 clients on each device, and an average delay was calculated as described in

$$\overline{\text{AD}} = \sum_{n=1}^{N} D_{\text{ete}}, \tag{2}$$

where $\overline{\text{AD}}$ is average delay experienced by 1,000 clients, D_{ete} end-to-end delay of the n^{th} client, and $N = 1{,}000$ clients.

The simulation is implemented for 1000 nodes, and the payload size is kept variable; the minimum payload size is 100 bytes while the maximum payload size used is 250 bytes. The nodes kept connecting and terminating the server after publishing the data on the server. For the ease of working and to make simulation more realistic, 25 nodes are connected to the sink node simultaneously and are allowed to publish the sensor data on the sink node which is then published to the MQTT server. All nodes are then disconnected from the sink node after data is published. The simulation parameter values are listed in Table 2.

5. Results and Discussion

To evaluate the system, the following use cases were used: sensor node with temperature and humidity sensor: the sensor node with temperature and humidity sensor module DHT-11 is deployed using Arduino and Xbee communication module, and several readings for the temperature and humidity are taken, and then, the readings are plotted in a graph; sensor node with CO_2 and methane gas sensor: the sensor node with CO_2 gas and methane gas detector, the MQ-135 gas sensor is deployed using Arduino and Xbee communication module, and several readings for the presence of CO_2 gas and methane gas are taken, and then, a graph was plotted; and sensor node with moisture sensor: the sensor node with soil moisture sensor HL-69 is deployed using Arduino and Xbee communication module to take several moisture readings, and a graph was plotted.

Readings were recorded from different sensor modules placed at different locations in the experimental area. The data is recorded after an interval of 1 hour, and the graphs were plotted. In Figure 8, the blue line shows the temperature reading taken from the sensor node 1 placed in room A of the experimental area. The temperature is taken and is measured in the Celsius scale. The maximum value recorded was 27.2 degrees while average temperature recorded was about 26.72 degrees. The silver line shows the temperature reading taken from sensor node 2 placed in room G near a gas stove. The maximum value recorded was 32.6 degrees while average temperature recorded was about 29.21 degrees. The orange line shows the humidity value recorded from sensor node 1 placed in room A. As the sensor supports for both temperature and humidity, the values can be retrieved from the sensor by sending the request for the specific value. In our case, the sensor retrieves the value of the temperature, if it receives a request containing “t” and sends the value for the humidity if the receiving request contains “h.” The maximum value was 98% average humidity value recorded was 74.9%. The yellow line shows the moisture reading taken from the sensor node 4 placed in open area B. The moisture is taken and is measured in the 10-bit ADC. The maximum value recorded was 800 while average moisture value recorded was about 479.63.

Figure 9 presents the readings related to methane and CO_2 gas recorded from the sensor node 3 placed in room G near the gas stove with gas supply on. The values of the methane gas and CO_2 gas were recorded in the parts per million (ppm) unit. The maximum values recorded for methane and CO_2 were 475 ppm and 140 ppm, respectively, while average readings recorded for the methane gas and CO_2 gas were 202.5 ppm and 90.29 ppm, respectively.

Following graphs show the result of simulation performed using OMNET++ simulator and Wireshark. Figure 10 presents the delay experienced in connection time with respect to the number of devices. As it is shown, average delay in connection increases with respect to the number of devices. By increasing the number of devices as well as the number of clients, load on server increases because of which every client experienced delay in its connection time. Similarly, minimum connection time increased linearly.

Figure 11 represents the average response time of the MQTT server. It is observed that average response time becomes largely linear with the number of devices. By making a small change in number of devices, clients experienced larger delay in response time.

Data shown in Figure 12 is the mean end-to-end delay recorded in seconds. The delay is higher for QoS-0 as it is the simpler level and no acknowledgement is delivered for any data either published or subscribed.

Similarly, Figure 13 shows the mean message loss in the network. It is seen clearly that the message loss in QoS-2 is less as compared to that in QoS-0 and QoS-1.

From the results, it is clear that the proposed system can be used to develop an IoT-based smart city and provide different facilities using the IoT services. Moreover, the MQTT protocol that is used for the development of the system is better than the SIP protocol because of the following reasons: (1) MQTT is a lightweight protocol than SIP; (2) MQTT provides a very light header of just 2 bytes but is also capable of providing a flexible header size of up to 256 bytes making it suitable for handling video transmission over the network using green MQTT; (3) the MQTT is a published/subscription-based network where SIP is a request-/response-based network; hence, MQTT handles requests efficiently than SIP; (4) MQTT also supports message payload up to 1000 bytes and makes the packet size relatively easier to handle; and (5) the average end-to-end delay and message loss are relatively less in the QoS-2 level, and it is clear that more a optimized form of the network can support more number of devices with less failure.

The proposed model is also a cost-effective model in terms of sensor node design as it provides a low-cost sensor node. Also, using a sink node common for several sensor nodes (1 sink node for at least 150 sensor nodes) helps to reduce the cost of the network deployment. The proposed model is also good for implementation in the countries, like Pakistan, as it is suitable for the weather condition as that of Pakistan.

6. Conclusion

The model proposed in this research is cost-effective and adaptive by the addition of many other services and firm technological support. The proposed model is simple and easy to implement with simple technology. The working and data collection and sharing are easy as it uses a simple way of communication. Moreover, as data is sent periodically between the sensor node and sink and sink and IoT cloud, thus the unnecessary overhead on the cloud as well as on the sink is removed. The data can also be fetched using local Bluetooth connection so not every user needs to connect with the IoT cloud. Third, the data can be fetched on demand; hence, new values can also be available even if the periodic cycle is not complete. The network is capable of handling more than 2000 devices in a single scenario with minimum delay and acceptable performance and efficiency. It is because of the stage-wise communication of the system. This model, at its infant stage, can be implemented at many places in Pakistan, even at a small-scale level, i.e., house, offices, or in small industrial areas. Furthermore, these small-scale projects also help the concept of smart urbanization to get fame and acceptance by the people of Pakistan. As a future prospective of the proposed model, a smart mobile-based package can be presented that supports connectivity of smart mobile devices. Besides, the factors regarding security, flexibility, scalability, and mobility can be addressed. For reducing the connectivity of the devices with the network and to mitigate the linear affect in connectivity by increasing the device number, the proposed design can be modified to introduce the factor mobility in the sink node. Another modification can be done by increasing the number of sink nodes twice relative to the sensor nodes.

Data Availability

All the necessary data is available in this manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors are grateful to the Taif University Researchers Supporting Project (number TURSP-2020/36), Taif University, Taif, Saudi Arabia. The authors are also grateful to the Deanship of Scientific Research at King Saud University for funding this research. This research work was also partially supported by the Faculty of Computer Science and Information Technology, University of Malaya, under Postgraduate Research Grant (PG035-2016A).

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: a survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] I. Ganchev, Z. Ji, and M. O'Droma, *A Generic IoT Architecture for Smart Cities*, 2014.

[3] L. Srivastava, *Japan's Ubiquitous Mobile Information Society*, info, 2004.

[4] S. Giroux and H. Pigot, *From Smart Homes to Smart Care: ICOST 2005, 3rd International Conference on Smart Homes and Health Telematics*, vol. 15, IOS Press, 2005.

[5] F. Xia, L. T. Yang, L. Wang, and A. Vinel, "Internet of Things," *International Journal of Communication Systems*, vol. 25, no. 9, pp. 1101-1102, 2012.

[6] M. A. Chaqfeh and N. Mohamed, "Challenges in middleware solutions for the internet of things," in *2012 International Conference on Collaboration Technologies and Systems (CTS)*, Denver, CO, USA, May 2012.

[7] M. Gigli and S. G. Koo, "Internet of Things: services and applications categorization," *Advances in Internet of Things*, vol. 1, no. 2, pp. 27–31, 2011.

[8] S. Jain, S. Mane, J. Lopez et al., "A low-cost custom HF RFID system for hand washing compliance monitoring," in *2009 IEEE 8th International Conference on ASIC*, Changsha, China, October 2009.

[9] J. Jin, J. Gubbi, S. Marusic, and M. Palaniswami, "An information framework for creating a smart city through Internet of Things," *IEEE Internet of Things Journal*, vol. 1, no. 2, pp. 112–121, 2014.

[10] N. Koshizuka and K. Sakamura, "Ubiquitous ID: standards for ubiquitous computing and the Internet of Things," *IEEE Pervasive Computing*, vol. 9, no. 4, pp. 98–101, 2010.

[11] R. S. Kshetrimayum, "An introduction to UWB communication systems," *IEEE Potentials*, vol. 28, no. 2, pp. 9–13, 2009.

[12] N. Kushalnagar, G. Montenegro, and C. Schumacher, *IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals*, 2007.

[13] Y. Leng and L. Zhao, "Novel design of intelligent Internet-of-Vehicles management system based on cloud-computing and Internet-of-Things," in *Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology*, Harbin, China, August 2011.

[14] P. Levis, S. Madden, J. Polastre et al., "TinyOS: an operating system for sensor networks," in *Ambient intelligence*, pp. 115–148, Springer, 2005.

[15] P. McDermott-Wells, "Bluetooth scatternet models," *IEEE Potentials*, vol. 23, no. 5, pp. 36–39, 2004.

[16] G. Montenegro et al., "Transmission of IPv6 packets over IEEE 802.15. 4 networks," *Internet proposed standard RFC*, vol. 4944, p. 130, 2007.

[17] U. Rushden, *Belkin Brings Your Home to Your Fingertips with WeMo Home Automation System*, Press Room Belkin, 2012.

[18] I. Ungurean, N.-C. Gaitan, and V. G. Gaitan, "An IoT architecture for things from industrial environment," in *2014 10th International Conference on Communications (COMM)*, Bucharest, Romania, May 2014.

[19] C. Wang, Z. Bi, and L. da Xu, "IoT and cloud computing in automation of assembly modeling systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1426–1434, 2014.

[20] R. Want, "An introduction to RFID technology," *IEEE Pervasive Computing*, vol. 5, no. 1, pp. 25–33, 2006.

[21] R. Want, "Near field communication," *IEEE Pervasive Computing*, vol. 10, no. 3, pp. 4–7, 2011.

[22] X. Xiaojiang, W. Jianli, and L. Mingdong, "Services and key technologies of the Internet of Things," *ZTE Communications*, vol. 8, no. 2, pp. 26–29, 2020.

[23] P. Bellavista, G. Cardone, A. Corradi, and L. Foschini, "Convergence of MANET and WSN in IoT urban scenarios," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3558–3567, 2013.

[24] H. Schaffers, N. Komninos, M. Pallot, B. Trousse, M. Nilsson, and A. Oliveira, "Smart cities and the future internet: towards cooperation frameworks for open innovation," in *The future internet assembly*, Springer, Berlin, Heidelberg, 2011.

[25] D. Evans, "The internet of things: how the next evolution of the internet is changing everything," *CISCO White Paper*, vol. 1, no. 2011, pp. 1–11, 2011.

[26] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, 2014.

[27] I. Vilajosana, J. Llosa, B. Martinez, M. Domingo-Prieto, A. Angles, and X. Vilajosana, "Bootstrapping smart cities through a self-sustainable model based on big data flows," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 128–134, 2013.

[28] J. M. Hernández-Muñoz, J. B. Vercher, L. Muñoz et al., "Smart cities at the forefront of the future internet," in *Future internet assembly*, 2011.

[29] E. A. Mulligan and M. Olsson, "Architectural implications of smart city business models: an evolutionary perspective," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 80–85, 2013.

[30] N. Walravens and P. Ballon, "Platform business models for smart cities: from control and value to governance and public value," *IEEE Communications Magazine*, vol. 51, no. 6, pp. 72–79, 2013.

[31] P. Masek, J. Hosek, K. Zeman et al., "Implementation of true IoT vision: survey on enabling protocols and hands-on experience," *International Journal of Distributed Sensor Networks*, vol. 12, no. 4, 2016.

[32] P. Masek et al., "Use case study on embedded systems serving as smart home gateways," *Recent Advances in Circuits, Systems and Automatic Control*, pp. 310–315, 2013.

[33] W. T. Muswera and A. Terzoli, *Development of an IMS Compliant, Cross Platform Client Using the JAIN SIP Applet Phone*, Development of an IMS Compliant, Cross Platform Client Using the JAIN SIP Applet Phone, 2010.

[34] S. Zapolskytė and V. Palevičius, "Overview and analysis of smart cities," *Mokslas–Lietuvos ateitis/Science–Future of Lithuania*, vol. 10, 2018.

[35] J. Lee, S. Baik, and C. Choonhwa Lee, "Building an integrated service management platform for ubiquitous cities," *Computer*, vol. 44, no. 6, pp. 56–63, 2011.

[36] V. Buscher and L. Doody, "Global innovators: international case studies on smart cities," *BIS Research Paper*, vol. 135, 2013.

[37] K. Ergazakis, K. Metaxiotis, and J. Psarras, "Towards knowledge cities: conceptual analysis and success stories," *Journal of Knowledge Management*, vol. 8, no. 5, pp. 5–15, 2004.

[38] R. R. Harmon, E. G. Castro-Leon, and S. Bhide, "Smart cities and the Internet of Things," in *2015 Portland International Conference on Management of Engineering and Technology (PICMET)*, Portland, OR, USA, August 2015.

[39] L. Laursen, *Barcelona's Smart City Ecosystem*, Cities Get Smarter, 2014.

[40] L. Bassett, *Introduction to JavaScript Object Notation: A To-the-Point Guide to JSON*, O'Reilly Media, Inc., 2015.

[41] T. Bray, "The javascript object notation (json) data interchange format (No. RFC 8259)," Technical Report, 2017.

[42] C. Ortega-Corral, L. E. Palafox, J. A. García-Macías, J. Sánchez-García, and L. Aguilar, "End-to-end message exchange in a deployable marine environment hierarchical wireless sensor network," *International Journal of Distributed Sensor Networks*, vol. 10, no. 1, Article ID 950973, 2014.

[43] F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte, and D. Vrgoč, "Foundations of JSON schema," in *Proceedings of the 25th International Conference on World Wide Web*, Montréal, Québec, Canada, April 2016.

[44] B. Smith, "Introducing JSON," in *Beginning JSON*, pp. 37–47, Springer, 2015.