applied sciences

MDPI

Article

Securing Internet of Things Applications Using Software-Defined Network-Aided Group Key Management with a Modified One-Way Function Tree

Antony Taurshia [1], Jaspher W. Kathrine [1], J. Andrew [2,*] and Jennifer Eunice R [3,*]

[1] Division of Computer Science and Engineering, Karunya Institute of Technology and Sciences, Coimbatore 641114, Tamil Nadu, India; antony18@karunya.edu.in (A.T.); kathrine@karunya.edu (J.W.K.)
[2] Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, Karnataka, India
[3] Department of Mechatronics Engineering, Manipal Institute of Technology, Manipal Academy of Higher Education, Manipal 576104, Karnataka, India
* Correspondence: andrew.j@manipal.edu (J.A.); jennifer.r@manipal.edu (J.E.R.)

Abstract: Group management is practiced to deploy access control and to ease multicast and broadcast communication. However, the devices that constitute the Internet of Things (IoT) are resource-constrained, and the network of IoT is heterogeneous with variable topologies interconnected. Hence, to tackle heterogeneity, SDN-aided centralized group management as a service framework is proposed to provide a global network perspective and administration. Group management as a service includes a group key management function, which can be either centralized or decentralized. Decentralized approaches use complex cryptographic primitives, making centralized techniques the optimal option for the IoT ecosystem. It is also necessary to use a safe, scalable approach that addresses dynamic membership changes with minimal overhead to provide a centralized group key management service. A group key management strategy called a one-way Function Tree (OFT) was put forth to lower communication costs in sizable dynamic groups. The technique, however, is vulnerable to collusion attacks in which an appending and withdrawing device colludes and conspires to obtain unauthorized keys for an unauthorized timeline. Several collusion-deprived improvements to the OFT method are suggested; however, they come at an increased cost for both communication and computation. The Modified One-Way Function Tree (MOFT), a novel technique, is suggested in this proposed work. The collusion resistance of the proposed MOFT system was demonstrated via security analysis. According to performance studies, MOFT lowers communication costs when compared to the original OFT scheme. In comparison to the OFT's collusion-deprived upgrades, the computation cost is smaller.

Keywords: Internet of Things; key management; group management; security; software-defined networks

Citation: Taurshia, A.; Kathrine, J.W.; Andrew, J.; Eunice R, J. Securing Internet of Things Applications Using Software-Defined Network-Aided Group Key Management with a Modified One-Way Function Tree. *Appl. Sci.* **2024**, *14*, 2405. https://doi.org/10.3390/app14062405

Academic Editor: Grzegorz Kołaczek

Received: 21 February 2024
Revised: 3 March 2024
Accepted: 5 March 2024
Published: 13 March 2024

1. Introduction

The Internet of Things (IoT) emerged when everyday objects began connecting to the Internet and interacting with each other autonomously without human intervention. IoT devices follow the IEEE 802.15.4 standard [1] to enable network connectivity for resource-constrained devices using short-range, lightweight communication protocols [1]. Due to the limited availability of device resources in addition to the bandwidth problem, the IoT ecosystem prefers multicast communication instead of unicast messages to send updates and patch-ups [2,3]. To send multicast messages securely, a common secret key need to be shared between the devices in the multicast group. To efficiently distribute the keys to all group devices, a vast amount of group key management techniques have been proposed in the literature. Group key management enables the group to operate with

integrity and confidentiality [4]. There are three methods for managing group keys. The Key Management Server (KMS), a trustworthy third party, is utilized for key distribution in the centralized key management technique [5,6]. In decentralized key management, both the server and group member devices contribute to key management [7–9]. In distributed key management, there is no centralized trust, but every member participates in key management [10–13]. Among all three methods, centralized key management offers less communication and computation overhead on the member devices with simpler functions. Hence, the proposed work focuses on centralized key management techniques, considering them to be most suitable for resource-constrained IoT devices. Centralized key management relies on a dependable third-party server for key management and key distribution. The proposed group management server uses an SDN controller to obtain centralized control over the heterogeneous network. The controller provides opinions on whether to forward network traffic, while the routers just obey the controller [14]. The advantages of SDN in comparison with traditional networks [15–17] are,

- Easy patching and upgradation
- Knowledge of the sleep/wake cycle of IoT devices
- Supports security services by routing traffic through virtualized service functions known as Virtual Service Functions (VSF)

The distribution of a common group key or key materials needed for group key generation occurs during the rekeying procedure. Rekeying ensures the group's confidentiality by putting forward and backward secrecy into action. When a group member leaves, the departing member should not have any access to any key materials that could be used to access any unapproved group data after the member exit event. This is termed forward secrecy. Similarly, when a new member enters an existing group, the joining member ought not to receive any key materials to obtain any unauthorized group data preceding the member join event. This is termed backward secrecy. A collusion attack is when two or more members join to obtain key material to access the group's data that are unauthorized for the colluding members. An efficient key management technique should ensure both forward secrecy and backward secrecy, as well as resistance to collusion attacks. Chinese Remainder Theorem (CRT)-based methods [18] offer the lowest communication costs out of all the centralized group key management techniques that have been proposed in the literature. However, because of its limited scalability, the method cannot be applied to dynamic groups.

A group key management strategy called OFT [19] lowers the communication cost for rekeying by a factor of $\log_2 n$. However, a collusion attack is possible on this approach. There are many proposed collusion-resistant OFT-based methods; however, they come at a higher communication cost. In the proposed MOFT, the binary tree's neighboring nodes share a second key, which is termed adjacent secret. Additionally, the method employs both a top-down and bottom-up strategy for group key generation, making it collusion-resistant with lower communication costs than the original OFT scheme. The suggested approach's performance analysis demonstrates that it has less communication overhead and better computation overhead than collusion-resistant OFTs. The existence of forward and backward secrecy in the MOFT approach is verified via security analysis. The existence of collusion resistance in MOFT is verified by a new proposition. Our main contributions are as follows:

- ➢ A novel technique, MOFT, for centralized group key management.
- ➢ A new proposition that proves the collusion resistance property of MOFT.
- ➢ The evaluation of MOFT proves MOFT reduces network traffic, with limited storage cost and optimal computation cost, proving it is scalable.

The remainder of the manuscript is organized as follows: Section 2 depicts the works that are related to SDN-based security for IoT and group key management techniques; Section 3 describes the proposed framework for group management using SDN, provides a brief on the original OFT scheme, and finally, elucidates on the proposed MOFT scheme;

Section 4 verifies the security strength of the proposed approach; Section 5 evaluates the proposed approach in terms of computation, storage, and communication cost; finally, Section 6 depicts the IoT tailored version of MOFT; and Section 7 gives the conclusion.

2. Related Works

2.1. SDN-Centered Security for IoT

An architecture for furnishing network security services like an internet content filtering system, firewall, Intrusion Detection System (IDS), Intrusion Prevention System (IPS), and packet inspection using SDN is proposed. The network traffic can be routed through the services based on the needs of the user. The use of SDN for IoT networks to mitigate attacks by limiting the rate of suspicious traffic flow is proposed in [20]. An SDN-based framework for designing cyber resilience for the Industrial IoT (IIoT) is proposed in [21]. The ontology is designed to create pre-programmed failover paths that become activated in the event of failure. This maintains the equilibrium resilience of the IIoT network for effective failure recovery. A similar SDN-based ontology design for cyber resilience in smart manufacturing applications is proposed in [22]. To implement incident response in the IIoT, the use of SDN is proposed in [23] due to its dynamic routing policies. The preconfigured incident–response policy is enforced using SDN in the event of an attack. Invariant-based anomaly detection using SDN for IIoT is proposed in [24]. Invariant is a property of the IIoT network that remains the same in any situation. A change in this property is identified as an anomaly by adding the invariant algorithm to the SDN-controlled switches. An Intrusion Detection and Prevention System (IDPS) is proposed in [25] that exploits SDN and a Genetic Algorithm (GA) to extract features for effective intrusion detection in IoT applications. Advanced reservation-based access control using SDN is proposed in [26]. The advanced reservation of bandwidth for a certain period is employed using SDN. SDN extends the reservation from border routers to the end devices with tokens for authorization. Another SDN-based architecture for smart home networks is proposed in [27]. With SDN, all smart home devices are connected via a gateway. KNOT and Orchestrator are used with the aid of SDN to detect Advanced Persistent Threats and saturation attacks and mitigate them effectively. HanGaurd SDN-based fine-grained protection for a smart home from malicious apps running on authorized devices like a smartphone is proposed in [28]. An SDN-based firewall called FORTRESS is proposed in [29]. The stateful flow data are obtained from the data plane, and the Mealy machine is exploited to perform state table updations based on the routing decision made. An SDN-centered defense mechanism for IoT networks is proposed in [30]. The IoT devices are classified as easily patchable or non-patchable, vulnerable or hard-to-exploit, and a proactive defense mechanism is provided by changing the attack surface in case the device is vulnerable and non-patchable. The features of SDN are exploited to provide a honeypot as a service by steering the traffic through Virtualized Functions (VF) as proposed in [31]. The honeypot acts as a proactive as well as a reactive defense mechanism against attacks. The honeypots are virtualized and provided as a service using SDN. SDN for effective intrusion detection as well as for mitigation of attacks like Distributed Denial of Service (DDoS) in the habitat of the IoT is proposed in [32,33]. DDoS detection system that exploits the convenience of SDN using machine learning techniques is proposed in [34]. The proposed work uses an adaptive multilayered feed-forwarding scheme that uses different algorithms in five layers. The third layer computes the live, real-time network traffic for DDoS attack detection, and SDN mitigates the attacks using Open Flow switches.

2.2. Group Key Management Techniques

SKDC follows a very simple approach, where all the existing participants of the group share an individual secret key with the KMS. When there is a change in the number of group members, the new group key is handed out by KMS to existing participants of the group individually, encrypted using the shared secret key, in sequential order. Hence, the communication cost is linear, which is highly inefficient for large dynamic groups.

The group Diffie–Hellman (DH) proposed in [35] uses asymmetric encryption for key distribution, with higher computation overhead. Logical Key Hierarchy (LKH) [36] is a familiar centralized, hierarchy-based key management approach for dynamic groups. LKH uses a balanced binary tree-based data structure with member devices' keys as leaf nodes. The root node contains the group key. The intermediate nodes along the path of the member devices to the root hold the key encryption keys with which the devices can obtain the group key. The overall overhead of the approach is 2 log n.

Yet another centralized, hierarchy-based key management approach for groups is the OFT, which reduces the communication cost to log n. Hence, the approach reduces bandwidth consumption considerably. The approach is similar to LKH, with the member device's individual keys as leaf nodes. The root key is the group key. The intermediate nodes hold the key-encryption key, which is calculated using a one-way and mixing function on the child node's keys. However, the approach fails to provide collusion resistance where two expelled members collide and can gain access to unauthorized group data. Several advancements to the OFT-based approach are proposed [37,38] to enhance the collusion resistance property, but in turn, the advancements increase the overall overhead. To impart collusion resistance in OFT, two more approaches are proposed in [39]; both approaches successfully impart collusion resistance with the same communication cost as OFT but with higher computation costs.

A centralized key management approach for the group that uses Diffie-Hellman for generating the key-encryption key is proposed in [40]. The group key can be decrypted using the key-encryption key. However, the use of public key cryptography increases the computation cost greatly. Moreover, without authentication, the use of Diffie-Hellman is liable to man-in-the-middle attack. A lightweight key management approach for groups formed in IoT applications is proposed in. The approach uses a hybrid technique with a combination of CRT and LKH. The intermediate node keys are calculated by hashing the device's ID. Although the ID of a device is unique, tracking the ID of the device is simple for malicious users to perform forgery-based attacks. Group key management using the Chinese Reminder Theorem (CRT) is proposed in. The approach is efficient, with the least communication cost of a single broadcast. When the group size exceeds the preset n value, the individual keys for the entire group must be renewed. A blockchain-based solution for key management in groups for autonomous aerial vehicles has earlier been proposed. The approach uses LKH along with blockchain for group key updations and reduces delay. The approach also assures forward and backward secrecy. However, the use of blockchain in a resource-constrained environment would be a problem. Another lightweight asymmetric group key management approach for VANETS is proposed in [41]. The approach uses a combination of CRT and asymmetric cryptography in contrast to the traditional symmetric group key management techniques. The scheme is comparatively scalable and has minimal computation overhead compared to its predecessor, CRT-based group key management techniques for VANETS. Still, the approach cannot use variable key sizes since the key sizes are chosen based on constraints. Further, a collusion-resistant approach to group key management is proposed in [42], which makes use of tokens to avoid collusions. The approach distributes group keys with a single broadcast with minimal communication load. Still, the storage overhead is high as the devices store the tokens belonging to a device's cognate nodes. A novel protocol, GKMP, for key management in groups, is proposed in [43] to avoid collusion attacks during file sharing in the cloud. The scheme uses a group key generated by participants and not by a centralized cloud server, adding security in terms of file sharing. But the scheme uses RSA for key generation, which is expensive for IoT devices. Yet another communication-aware key management protocol for IoT networks is proposed in [44]. The scheme uses hyperelliptic curve cryptography for authentication along with bilateral generalization in a homogeneous short integer-based solution for effective key management in IoT groups. The scheme has lower computational time compared to its previous similar schemes; still, public key cryptography-based schemes increase the complexity and device overload in IoT networks.

When evaluated in terms of computation, communication, storage load, scalability, and secrecy, the existing techniques in the literature attain efficiency in one parameter with a trade-off in other parameters. OFT reduces the communication cost but does not ensure secrecy. CRT reduces the overall computation, communication, and storage costs but lacks scalability. Public-key cryptography-based schemes increase the computation load on the device to a greater extent. To provide group key management as a service, a centralized group key management technique should offer the following properties:

- ➢ Secrecy, ensuring forward secrecy, backward secrecy, and collusion resistance.
- ➢ Reduced communication costs, leading to reduced network traffic.
- ➢ Limited usage of resources in IoT devices.
- ➢ The technique should be scalable even though the group is large and dynamic.

3. Proposed Work

3.1. SDN-Based Group Management Framework

The architecture for group management service is depicted in Figure 1. The devices that communicate frequently and form a communication pattern are united as a group. The Group Management Server (GMS) is built-in with an authentication service function, a key management service function, and an SDN controller. Service functions are virtualized functions that can be provided as a service. The GMS stores the attributes of the group, including the group ID, group member ID, application ID, IP address, MAC address, and location. The key management function stores the key materials of the group. The authentication function authenticates the devices by collaborating with the GMS.

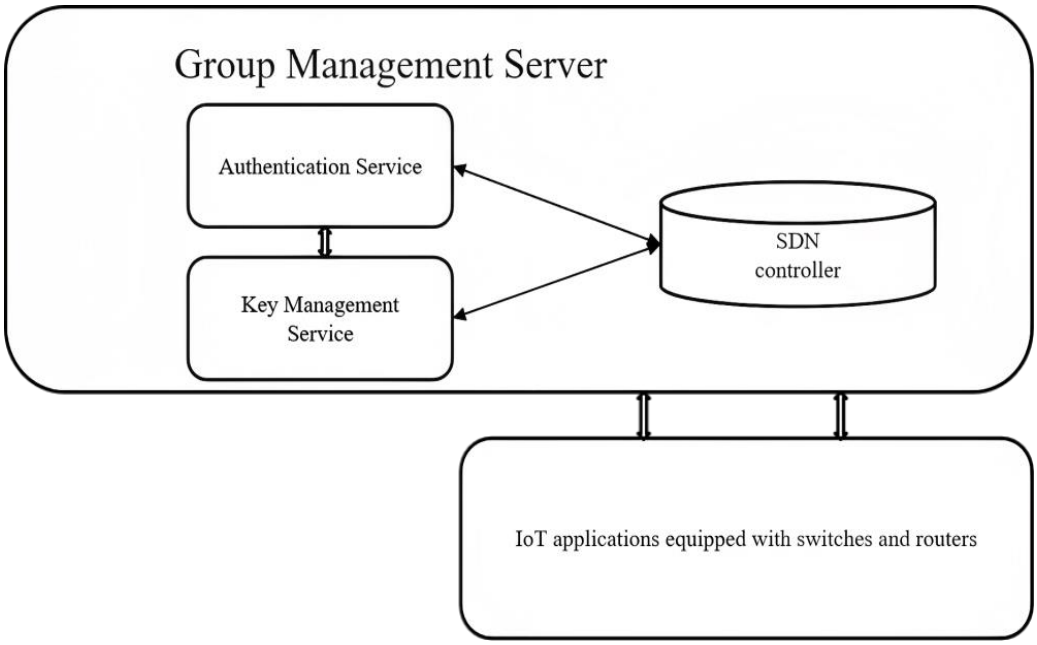


Figure 1. Group management server (GMS).

The authenticated devices will then be allowed to form a group. The key management service function will then distribute the key materials. Using SDN, the communication link is set up in such a way that only group members can communicate. This ensures security against the spread of node-capture attacks and phishing-based attacks by cutting off unnecessary communications.

3.2. Use Case

The proposed group management framework is applied to smart home applications, as depicted in Figure 2. The smart home application is built up of embedded devices and sensors that collaborate via LAN or WAN connections to improve the comfort of the house members. The components of a smart home system include an entertainment system consisting of a music system, home theater, TV, etc., a security surveillance system with cameras, alarms, door locks, etc., and an energy management system with a connection to a smart grid to optimize electricity usage [45,46]. The segregation of smart home applications into groups helps prevent the spread of attacks from a vulnerable part of a system to other parts of the system. An authenticated key agreement protocol for groups in home-based sensor networks is proposed in [47], which uses an authentication token for authentication between mobile stations and serving networks. The scheme shows robustness when the mobile stations move between different serving networks. Another group-based authenticated key agreement protocol for machine-to-machine communication is proposed in [48], which selects a leader device first to perform full authentication. The remaining member devices authenticate using the vector provided by the leader device. The proposed work assumes that the smart home application contains an admin device that is trustable with enough resources to communicate with service functions and establish a secure connection. The steps involved in group initialization are depicted in Figure 3. Resource-constrained smart home devices can connect with the admin device via 5G networks or Bluetooth LE that supports device authentication. After mutual authentication, the admin devices send the device IP, MAC address, ID, Group ID, individual key K_L, and adjacent shared secret K_a to the authentication service function encrypted using the key shared between them. After authentication, the authentication service function forwards the request to the key management service function. The key management service function forwards the adjacent shared secret and key materials needed to obtain the group key to the device encrypted using K_L, which was already obtained by the device from the admin. This ensures the message is from a trusted source. The member devices only know the group key of their respective groups. But the admin will have all the group keys as they generate K_a, which is consequently used to extract the group key in the proposed group key management approach. The flow rules are set in the switches and routers using an SDN controller so that only devices in the same groups communicate. Thus, if a malicious person compromises the music system, they cannot proceed further to compromise the door lock since the device will be in another group, and using SDN, these devices can be segmented.

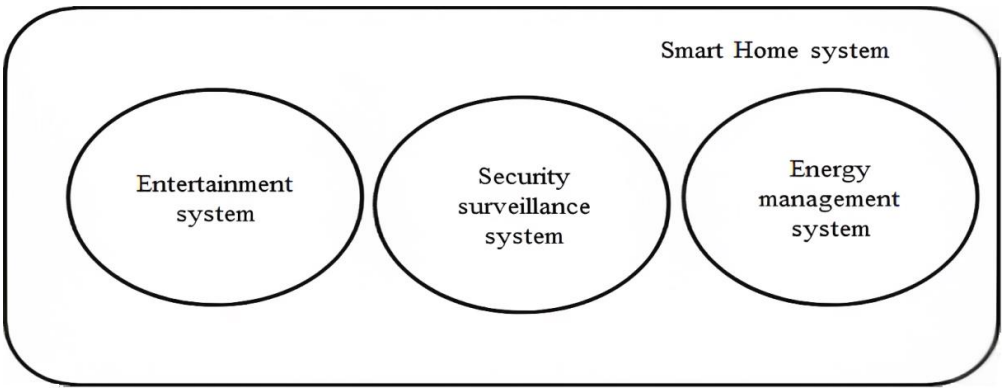


Figure 2. Grouping of the smart home system.

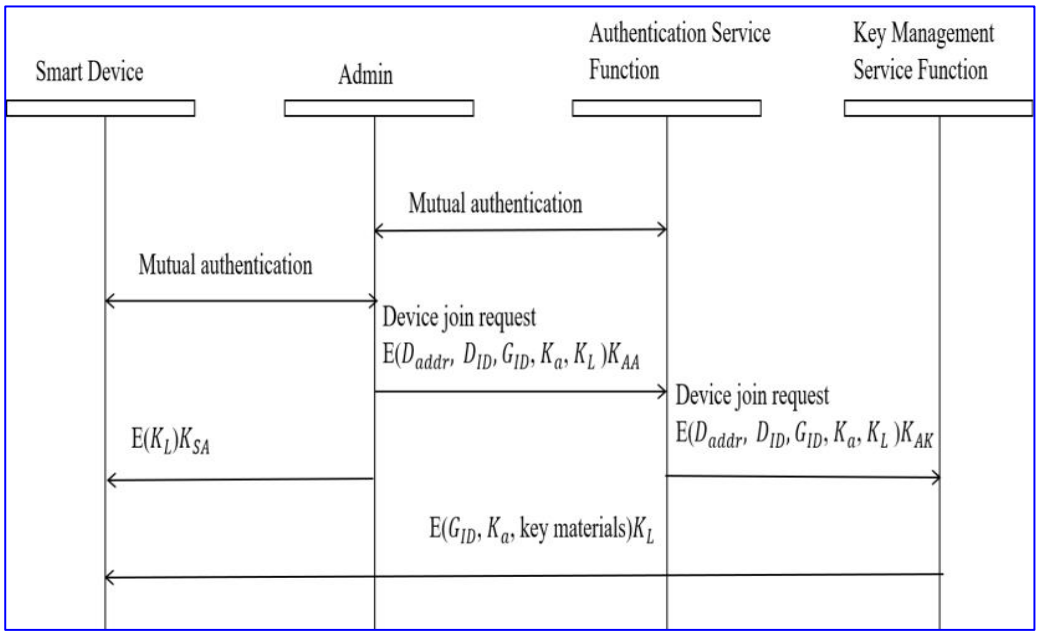


Figure 3. Steps in group initialization.

3.3. Existing OFT

Table 1 gives the notations used in the proposed work. The group key GK is commonly shared among all group devices to share data and resources within the group. The node key n_k is the individual key possessed by every node. An adjacent secret $K_a(l,r)$ is a secret key shared between two adjacent nodes, i.e., the left and right child of a parent node. The intermediate node is the node between the leaf and the root node that contains the group key. The intermediate node secret SI_i is the secret calculated and known only to the intermediate node, whereas the blinded secret bl_i is distributed to other nodes for group key generation.

Table 1. Notations used.

GK	Group key
n_k	node key
bl_i	Blinded secret of node i
I_i	Intermediate node
$K_a(l,r)$	Adjacent shared secret of node l and r
K_i	Individual secret key of user i
f(x)	One-way function
g(x)	Mixing function
L	Length of key
n	No. of users
SI_i	Node secret of Intermediate node i
K_L	Individual key between GMS and user
K_{AA}	Key shared between authentication service function and admin
K_{AK}	Key shared between authentication service function and key management service function
K_{SA}	Key shared between
G_{ID}	Group ID
D_{ID}	Device ID

Before looking into the working of MOFT, the working of the earliest OFT scheme is discussed. The OFT scheme uses a binary tree data structure, which is balanced for key management. Each leaf node of the binary tree denotes a group member and is assigned a node key. The group members already possess a shared secret with the KMS with which the

node key can be encrypted and distributed to the respective group member. The one-way function on the key is the blinded secret of the leaf node. The mixing function of the left and right child's blinded node secret gives the intermediate node secret. In this way, the intermediate node secrets are calculated in a bottom-up fashion to compute the group key.

In Figure 4, the numbers represent the nodes of the binary tree, whereas the intermediate nodes are represented using the symbol I. When a user exits or joins, the keys on the trail of the leaving or joining node are changed. As in Figure 5, when user 7 leaves, node 8 takes the position of I_3 as the leaf node. Node 3 is issued a new node key n_{k3}. Node 3 then calculates its new blinded node secret bl_3, thus leading to a new SI_1 intermediate node secret and group key GK. Users 9 and 10 are issued the new bl_3 encrypted using SI_4. Hence, users 9 and 10 calculate $SI_1 = g(bl_3, bl_4)$. Users 11 and 12 are issued new bl_{k1} encrypted using SI_2. Now, the group key is calculated as $GK = g(bl_1, bl_2)$.

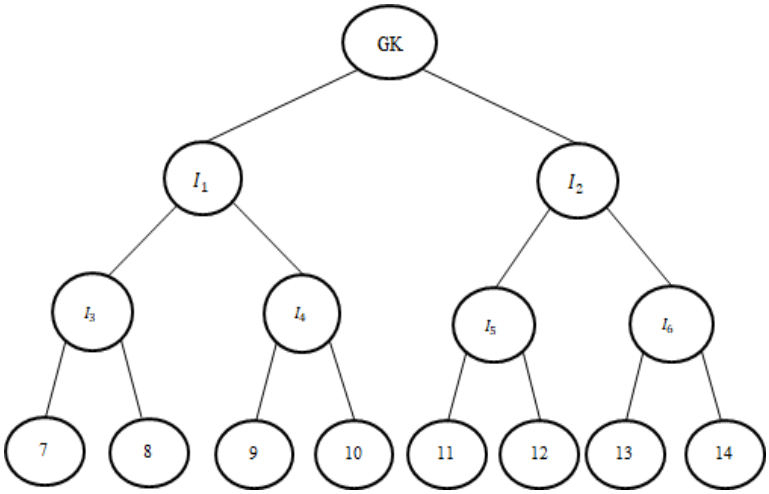


Figure 4. Group tree structure.

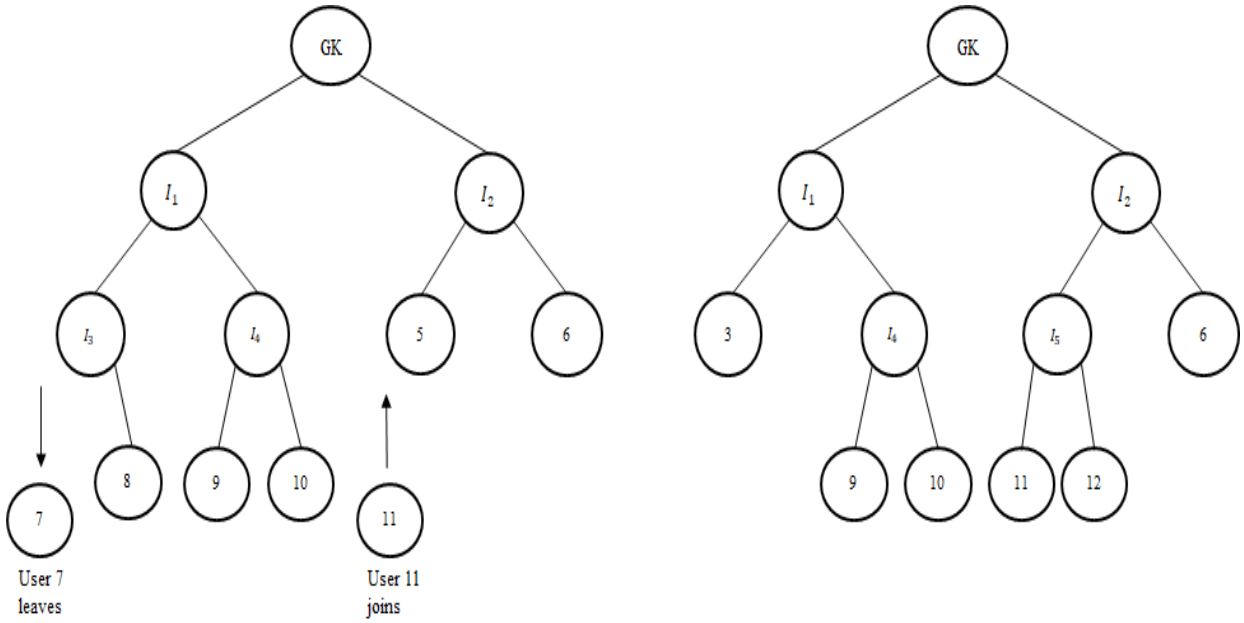


Figure 5. User leave and join event.

Now, as in Figure 5, if user 11 joins, the user is issued the node secret n_{k11}. User 11 will calculate its blinded secret bl_{11}. User 12 is issued bl_{11} encrypted using its own node secret n_{k12}. Users 11 and 12 calculate $= g(bl_{11}, bl_{12})$. Users 11, 12, and 6 calculate $SI_2 = g(bl_5, bl_6)$. Then, users 3, 9, and 10 are supplied with the blinded secret of node I_2 so that the group

key GK can be calculated. Although the scheme has the advantage of reduced communication costs, it is prone to collusion attacks. Several enhanced OFT-based approaches are proposed [38,39] to attain collusion resistance but with higher communication cost or computation cost.

Collusion Attack in OFT

When two users collude to obtain unauthorized information, it can be termed a collusion attack. For instance, when user 7 leaves at time t1, they know the bl_1 value. Now, new user 11 joins at time t2. User 11 knows the bl_1 value. Now, the left user 7 colludes with malicious user 11 and uses bl_2, which is known to user 7, and bl_1, which is known to user 11, to compute the group key $GK = g(bl_1, bl_2)$ for the timeline between t1 and t2. Thus, OFT is prone to collusion attacks. There are works based on OFT that avoid collusion attacks often but with an enhanced cost. Our work attempts to provide a collusion-resistant OFT-based scheme but with a reduced cost.

3.4. MOFT

The proposed MOFT scheme uses a binary tree just like OFT maintained by a key management server. The proposed scheme also assumes that the users joining the group have an individual secret key K pre-established with the KMS. In the modified OFT, instead of leaf nodes being issued with the node key, the adjacent leaf nodes are issued with a common adjacent shared secret $K_{a(l,r)}$.

Now, as in Figure 6, the left and right leaf nodes share a common adjacent shared secret $K_{a(l,r)}$. The first intermediate node nearest to the leaf nodes I_{i+2} will have the same $K_{a(l,r)}$ as node secret, and the blinded node secret of I_{i+2} is calculated as $f(K_{a(l,r)})$. The next intermediate node secret I_I is calculated as $g(bl_{i+2}, bl_{i+3})$. Further node secrets are calculated in a bottom-up style as in OFT to compute the group key GK.

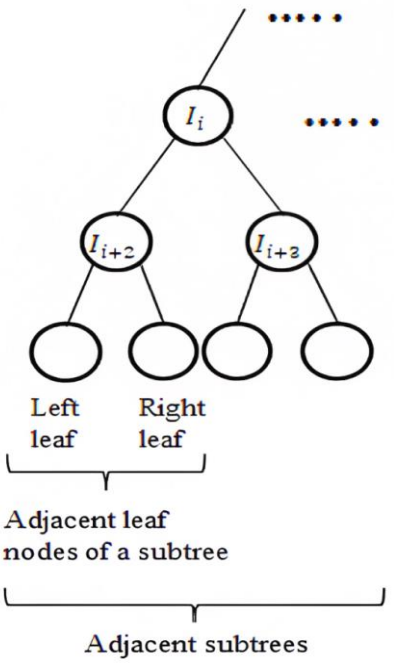


Figure 6. Adjacent subtree structure.

3.4.1. Top-Down Growth

As in Figure 7, when a new user u1 arrives, the l node performs a mixing function $g\left(K_{a(l,r)}, K_l\right)$ in which K_l is the pre-established secret of user l, and $K_{a(l,r)}$ is the adjacent shared secret between nodes l and r. This is the new adjacent secret for l and u1. The r node's blinded node secret becomes $f\left(K_{a(l,r)}\right)$. Similarly, when a new user arrives on the

right side of the tree, $g\left(K_{a(l,r)}, K_r\right)$ is calculated and issued to the new user as an adjacent secret. In this way, a certain amount of communication cost is cut off.

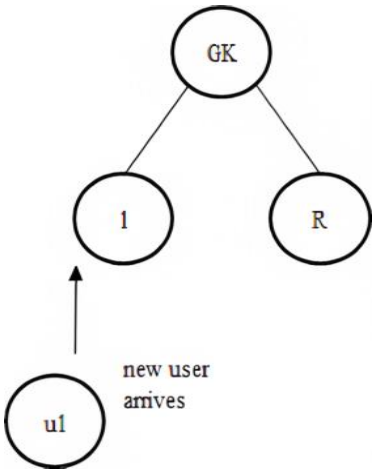


Figure 7. Top-down growth during new user join.

3.4.2. User Leave and Join Event

When user 7 leaves, as in Figure 8, node 8 becomes leaf node 3. In the modified OFT node, 3 is issued with a new adjacent shared secret value K_{a3}. The adjacent secret is blinded using a one-way function to obtain the blinded secret of node 3. Users 9 and 10 are supplied with a new blinded secret of node 3. SI_2 is calculated as $(g(bl_3, bl_4)$. Users 7, 8, 9, and 10 are supplied with the blinded secret of I_2. The keys that change during the user leave event are highlighted in Figure 8.

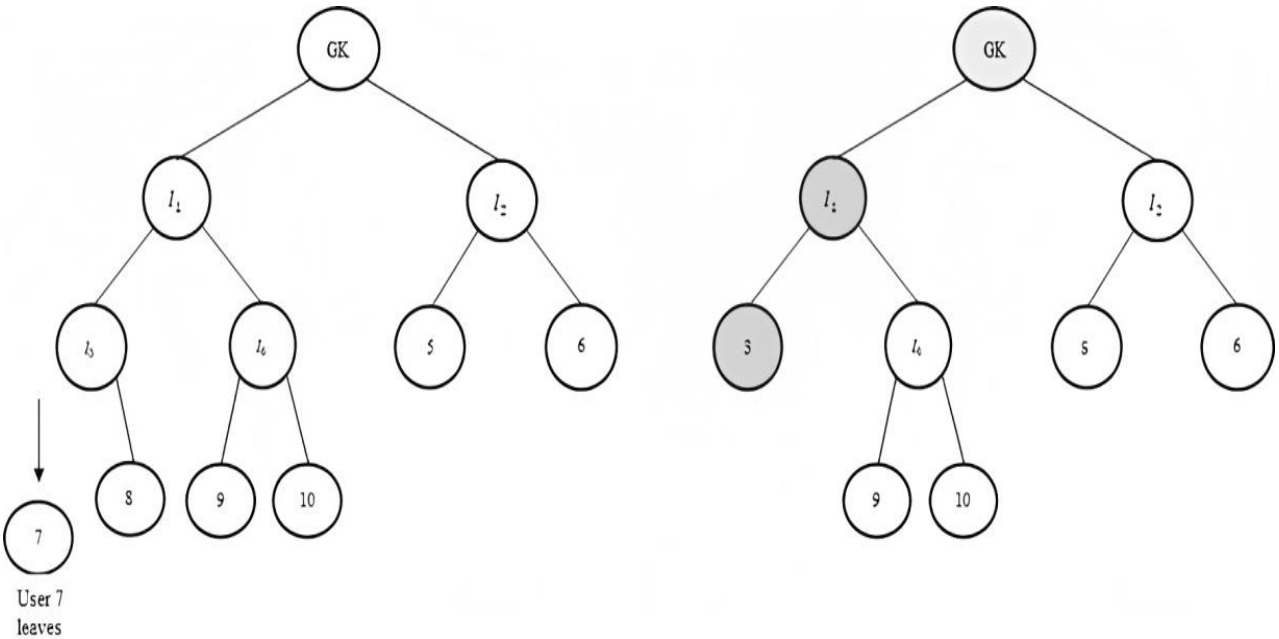


Figure 8. User leave event, keys that change during leave event.

When user 11 joins as in Figure 9, node 5's adjacent shared secret is changed to $g(K_{a(5,6)}, K_5)$. The changed adjacent secret is issued to user 11 as $K_{a(11,12)}$ encrypted using its individual secret key K_{11}. The blinded secret of I_5 is calculated as $f(K_{a(11,12)})$. User 6's adjacent secret $K_{a(5,6)}$ is blinded using a one-way function $f(x)$ and becomes its blinded secret $bl_6 = f(K_{a(5,6)})$. User 11 is issued with the blinded secret bl_6. Users 6, 11, and 12 calculate $SI_2 = g(bl_5, bl_6)$. Users 7, 8, 9, and 10 are supplied with SI_2, which is encrypted

using the node secret of I_1. As in ROFT [39], another one-way function is performed on each one of the blinded node secrets along the trail of the joining user and supplied to them as a new blinded secret. This makes the MOFT collusion-resistant. The keys that change during the user join event are highlighted in Figure 9. Hence, the new users entering the group do not know any previously used keys, assuring backward secrecy. The modified OFT reduces the communication cost compared to the original OFT.

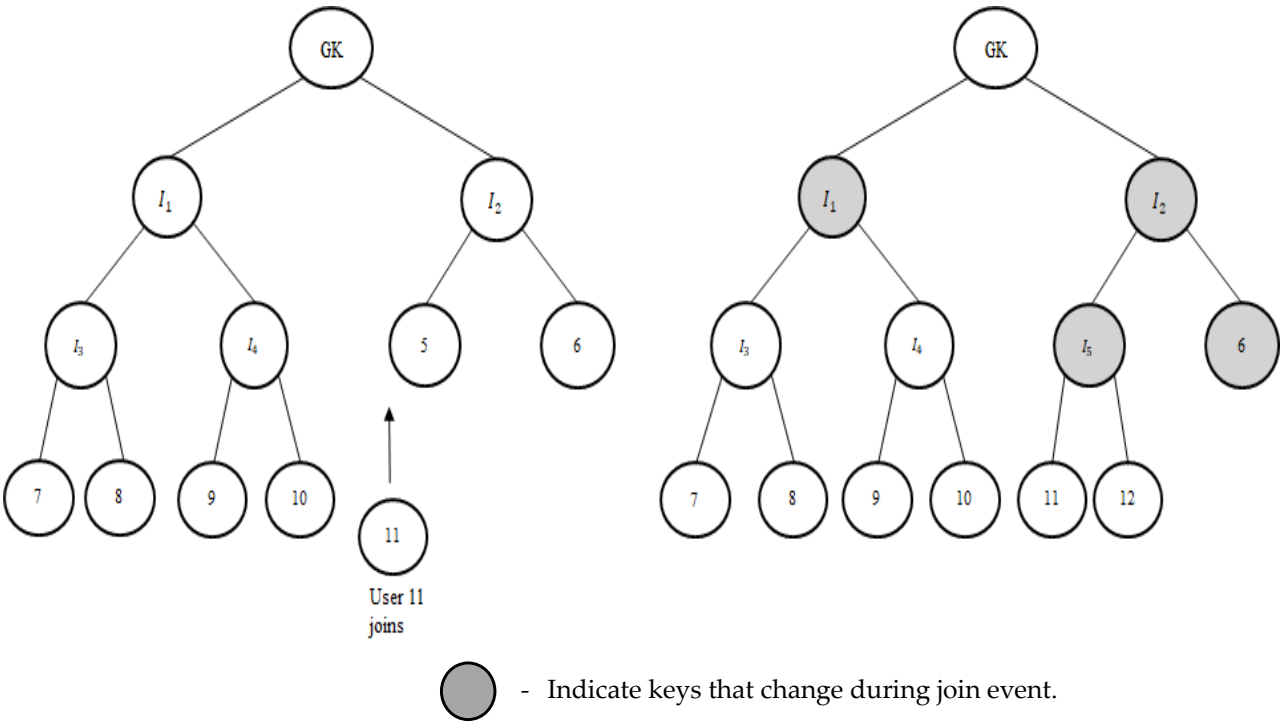


Figure 9. User join event.

4. Security Analysis

The work one way function tree (OFT) proposes a new generic definition for collusion resistance. Although our scheme follows ROFT to be collusion-resistant, a new proposition is made to prove the collusion resistance of our scheme.

Proposition 1. *The OFT scheme is collusion-resistant if the leaving users at the time t*1 *and joining users at time t*2 *cannot compute any unknown node secrets if the intersection of their set of known secrets does not contain any adjacent pair node secrets,* i.e., *the leaving users at time t*1 *known key set* t_{leave} *and the joining users at time t*2 *known key set* t_{join} *should not contain the secrets of adjacent pairs* $t_{leave} \cup t_{join} \neq secrets\ of\ pair\{I_{oi}, I_{oi+1}\}$ *where* $oi \in \{set\ of\ odd\ numbers\}$, t_{leave} *is the key set known in the time duration between t*1 *and the first key update after the user leaves* t_{MIN}, t_{join} *is the key set known in the time interval between the last key update that happened before the user join* t_{MAX} *and t*2.

Proof. The group key is calculated in a bottom-up fashion using a mixing function $g(x)$ of adjacent blinded node secrets bl_{oi}, bl_{oi+1}. So, to compute any parent intermediate node secret SI_i, both the child's adjacent blinded secrets and adjacent secrets should be known. Hence, if the chain of knowing adjacent secrets and adjacent blinded secrets is broken then the colluding users cannot compute any intermediate key or the group key for an unauthorized time. □

Theorem 1. *MOFT is collusion-resistant.*

Proof of Theorem. According to Proposition 1, the OFT scheme is collusion-resistant if the leaving users t_{leave} and the joining users t_{join} cannot collude to contain any pair $\{SI_{oi}, SI_{oi+1}\}$. Consider user Bob leaves at the time t_l, as in Figure 10 (B–E) subtree sections. The set of keys Bob knows after time t_l, $t_{leave} =$ secret key of nodes $\{R, R', R'', I_R\}$. To perform a collusion attack, Bob has to collude with any of the joining users at the time t_j. Consider Alice joining at the time t_j at any of the sections B, C, D, or E. The known key set of Alice that existed before time t_j, $t_{join} = \{null\}$ since all the node secrets on the path are hashed using the one-way function before being distributed to Alice. The collusion of Alice and Bob known key sets $t_{leave} \cup t_{join} =$ secret key of nodes $\{R, R', R'', I_R\}$. Hence, according to Proposition 1, the proposed MOFT confirms collusion resistance. □

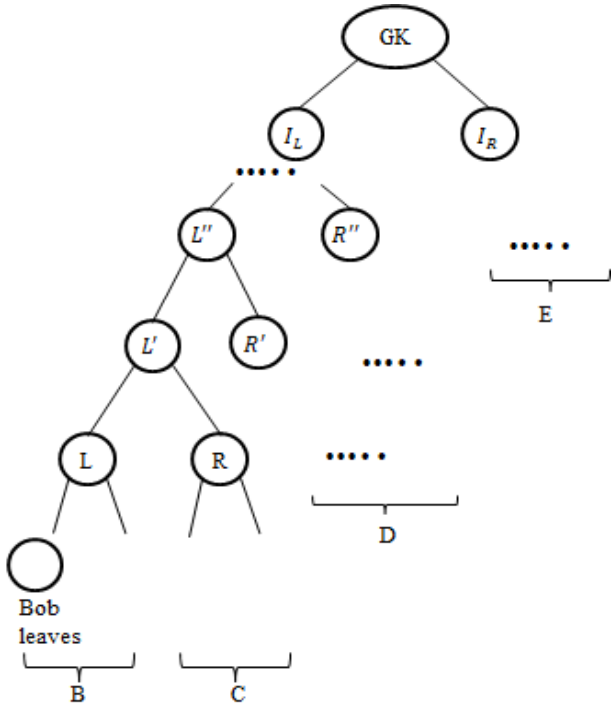


Figure 10. Collusion attack.

5. Performance Evaluation

The proposed MOFT is evaluated by comparing it with the prevailing OFT-based group key management approaches in terms of computation and communication cost.

From Tables 2 and 3, it is evident that the proposed MOFT approach attains the least communication cost, and the cost is even less than the OFT approach itself. The proposed MOFT reduces the communication cost by 39% than the earliest OFT. Although the computation cost is higher than the OFT scheme, it is comparatively less than the existing collusion-resistant OFT schemes. Figure 11 depicts the evaluation of the proposed approach with the existing approaches regarding total communication cost. The schemes OFT and ROFT share the same communication cost. Ku and Chen's and Xu's schemes share the same communication cost, whereas the proposed work's communication cost is less with $(2\log_2 n + 3) \times L$ messages, which is lesser than the original OFT. The number of messages transferred is equal to the number of keys needed to be distributed.

Table 2. User join event.

KMS	OFT [19]	Ku and Chen [37]	Xu [38]	ROFT [39]	MOFT
Comm. Cost KMS	$(2log_2n+1)\times L$	$(2log_2n+1)\times L$	$(2log_2n+1)\times L$ OR $((log_2n)^2+(2log_2n+1))\times L$	$(2log_2n+1)\times L$	$(log_2n+2)\times L$
Comp. Cost KMS	$(2log_2n+1)\times C_e+(log_2n+1)\times C_h$	$(2log_2n+1)\times C_e+(log_2n+1)\times C_h$	$(2log_2n+1)\times C_e+(log_2n+1)\times C_h$ OR $((log_2n)^2+2log_2n+1)\times C_e+((log_2n)^2+log_2n+1)\times C_h$	$(2log_2n+1)\times C_e+(2log_2n)\times C_h$	$(log_2n+2)\times C_e+(2log_2n-1)\times C_h$
Comp. Cost member device	$2C_d+log_2n\times C_h$	$2C_d+log_2n\times C_h$	$2C_d+log_2n\times C_h$ OR $(log2n+1)\times C_d+0.5\times(log_2n)\times C_h$	$2C_d+(2log_2n-1)\times C_h$	$C_d+(2log_2n-1)\times C_h$

Table 3. User leave event.

KMS	OFT [19]	Ku and Chen [37]	Xu [38]	ROFT [39]	MOFT
Comm. Cost KMS	$(log_2n+1)\times L$	$((log_2n)^2+log_2n+1)\times L$	$(log_2n+1)\times L$	$(log_2n+1)\times L$	$(log_2n+1)\times L$
Comp. Cost KMS	$(log_2n+1)\times C_e+(log_2n)\times C_h$	$((log_2n)^2+log_2n+1)\times C_e+((log_2n)^2+log_2n)\times C_h$	$(log_2n+1)\times C_e+(log_2n)\times C_h$	$(log_2n+1)\times C_e+(log_2n)\times C_h$	$(log_2n+1)\times C_e+(log_2n)\times C_h$
Comp. Cost member device	$C_d+log_2n\times C_h$	$(log_2n)\times C_d+(log_2n)^2\times C_h$	$C_d+log_2n\times C_h$	$C_d+log_2n\times C_h$	$C_d+log_2n\times C_h$

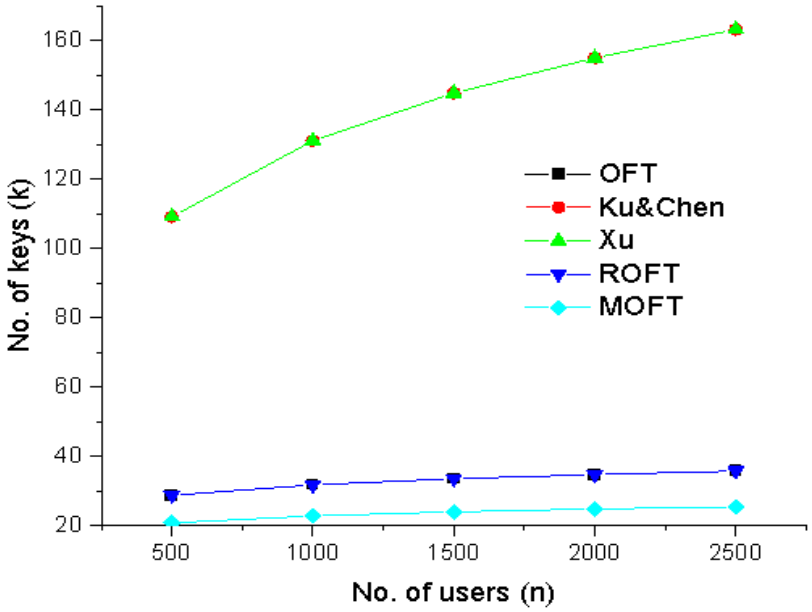
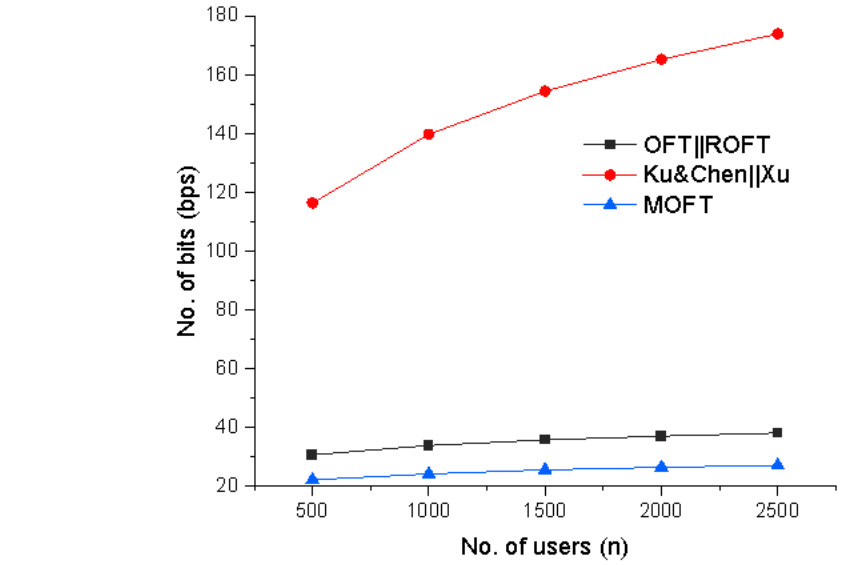


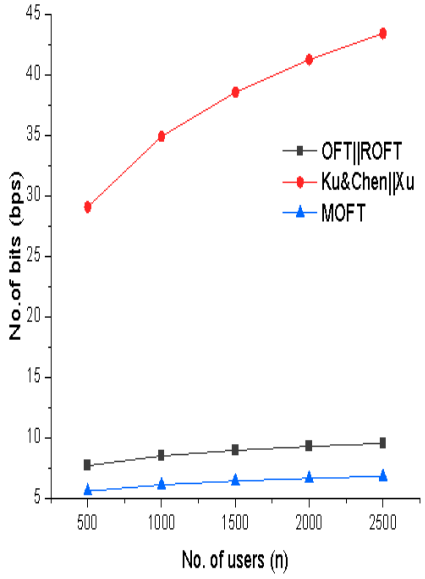
Figure 11. Comparison of proposed MOFT communication cost with OFT [19], Ku & Chen [37], Xu [38] and ROFT [39].

5.1. Load on SDN-Centered KMS

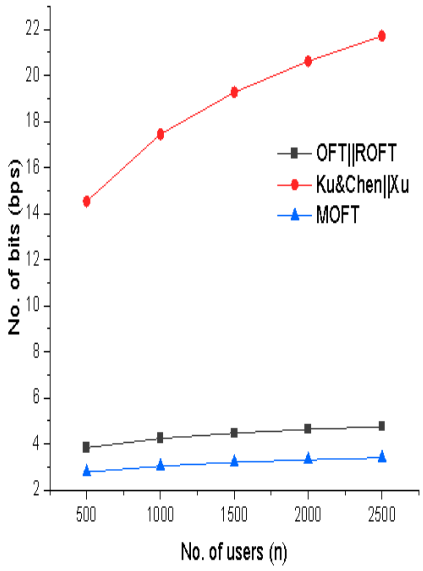
Group rekeying occurs in the event of the user joining or exiting the group. For static applications of IoT like smart home, the devices joining and leaving the smart home network are less frequent. For dynamic applications like retail, healthcare, etc., the rekeying frequency is high. Figure 12a–c depict the communication load on SDN-centered KMS for different rekeying frequencies f = $^1/_{120}$, $^1/_{480}$, and $^1/_{1920}$ respectively.



(**a**) Rekeying frequency $f = 1/120$



(**b**) Rekeying frequency $f = 1/480$



(**c**) Rekeying frequency $f = 1/1920$

Figure 12. Rekeying frequency comparison of proposed MOFT with OFT [19], Ku & Chen [37], Xu [38] and ROFT [39].

5.2. Storage Cost

The maximum storage cost of the OFT scheme is $2(\log_2 n + 1)$. The proposed work uses an adjacent secret shared between adjacent nodes instead of the individual node secret; hence, the storage cost of MOFT is $2\log_2 n$.

6. MOFT Tailored for IoT

Since the adjacent secret for the device is generated by the admin and shared with the centralized server, the admin can generate all the key materials subsequently needed to obtain the group key of all groups in the application. To reduce the overhead on the centralized key management server, the admin device can act as an edge device and take part in the key distribution part. The devices on the left side of the root node can obtain group keys from the centralized server, and the right-side devices can obtain their keys from the admin or vice versa. This will further reduce the overhead on the centralized server. To sum up, even though the group exhibits regular rekeying with membership changes,

the suggested MOFT minimizes the communication cost and thus reduces the network traffic to a significant extent. The proposed MOFT also offers lower storage overhead on group member devices. The suggested solution additionally provides collusion resistance using basic cryptographic primitives such as a one-way function and mixing function. The computation overhead is lower than the collusion-deprived enhancements of OFT. All these features make MOFT secure and adaptable to the IoT environment.

The proposed work using the SDN controller, switches, and routers are simulated using the Mininet [49] simulator installed in an oracle VM. The topology is created with 30 hosts, an SDN controller, and 2 switches. The flow rule is set using python script. The hosts are grouped into three groups with each group consisting of ten hosts. The communication link is set only for the hosts of the same group. This restricts the communication between hosts of different groups.

7. Conclusions

Unlike traditional networks, the IoT's network is heterogeneous, with different lightweight communication protocols for communication with lesser bandwidth. Treating this heterogeneous network as a single network and providing it with essential security solutions is inevitable. Hence, a group management framework with an SDN-centered server as a trusted third party is proposed to tackle heterogeneity and aid in secure key distribution. For key distribution to dynamic groups, a cost-efficient, scalable, and centralized key management technique is undeniable. Although CRT-based schemes are efficient, they lack scalability. LKH and OFT are the earliest, most familiar, and most widely used centralized key management techniques. OFT was proposed as an enhancement to LKH with reduced communication costs, yet it suffers from collusion attacks. The proposed MOFT technique is novel, has minimal communication overhead, and reduces network traffic by 39% compared to OFT. The new proposition defined for security analysis proves that MOFT is collusion-resistant with optimal computation overhead.

Author Contributions: Conceptualization, A.T. and J.W.K.; Methodology, A.T., J.W.K. and J.A.; Validation, J.E.R.; Writing—original draft, A.T.; Writing—review and editing, J.A. and J.E.R.; Visualization, J.E.R.; Supervision, J.W.K. and J.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used in this research are available upon request to the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Farash, M.S.; Turkanović, M.; Kumari, S.; Hölbl, M. An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment. *Ad Hoc Netw.* **2016**, *36*, 152–176. [CrossRef]
2. Heo, G.; Chae, K.; Doh, I. Hierarchical Blockchain-based Group and Group Key Management Scheme Exploiting Unmanned Aerial Vehicles for Urban Computing. *IEEE Access* **2022**, *10*, 27990–28003. [CrossRef]
3. Nikbakht Bideh, P. LMGROUP: A Lightweight Multicast Group Key Management for IoT Networks. In *International Conference on Information Security Practice and Experience*; Springer: Cham, Switzerland, 2022; pp. 213–230. [CrossRef]
4. Sakarindr, P.; Ansari, N. Survey of security services on group communications. *IET Inf. Secur.* **2010**, *4*, 258–272. [CrossRef]
5. Xu, J.; Li, L.; Lu, S.; Yin, H. A novel batch-based LKH tree balanced algorithm for group key management. *Sci. China Inf. Sci.* **2017**, *60*, 108301. [CrossRef]
6. Kung, Y.-H.; Hsiao, H.-C. GroupIt: Lightweight Group Key Management for Dynamic IoT Environments. *IEEE Internet Things J.* **2018**, *5*, 5155–5165. [CrossRef]
7. Kim, Y.; Perrig, A.; Tsudik, G. Tree-based group key agreement. *ACM Trans. Inf. Syst. Secur.* **2004**, *7*, 60–96. [CrossRef]
8. Zhou, W.; Xu, Y.; Wang, G. Distributed Group Key Management Using Multilinear Forms for Multi-privileged Group Communications. In Proceedings of the 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Melbourne, VIC, Australia, 16–18 July 2013. [CrossRef]

9. Sepulveda, J.; Flórez, D.; Immler, V.; Gogniat, G.; Sigl, G. Efficient security zones implementation through hierarchical group key management at NoC-based MPSoCs. *Microprocess. Microsyst.* **2017**, *50*, 164–174. [CrossRef]
10. Ali, S.; Rauf, A.; Islam, N.; Farman, H.; Jan, B.; Khan, M.; Ahmad, A. SGKMP: A scalable group key management protocol. *Sustain. Cities Soc.* **2018**, *39*, 37–42. [CrossRef]
11. De Salve, A.; Di Pietro, R.; Mori, P.; Ricci, L. Logical key hierarchy for groups management in Distributed Online Social Network. In Proceedings of the 2016 IEEE Symposium on Computers and Communication (ISCC), Messina, Italy, 27–30 June 2016; pp. 710–717. [CrossRef]
12. Inoue, D.; Kuroda, M. FDLKH: Fully decentralized key management scheme on logical key hierarchy. In *Applied Cryptography and Network Security*; Springer: Berlin/Heidelberg, Germany, 2004; Volume 3089, pp. 339–354. [CrossRef]
13. Wu, Q.; Qin, B.; Zhang, L.; Domingo-Ferrer, J.; Farras, O.; Manjon, J.A. Contributory broadcast encryption with efficient encryption and short ciphertexts. *IEEE Trans. Comput.* **2016**, *65*, 466–479. [CrossRef]
14. Der Chou, L.; Tseng, C.-W.; Huang, Y.-K.; Chen, K.-C.; Ou, T.-F.; Yen, C.-K. A Security Service on-demand Architecture in SDN. In Proceedings of the 2016 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Republic of Korea, 19–21 October 2016; pp. 287–291. [CrossRef]
15. Taurshia, A.; Kathrine, J.W.; Shibin, D. Prognostic Views on Software Defined Networks Based Security for Internet of Things. *Commun. Comput. Inf. Sci.* **2019**, *1116*, 100–116. [CrossRef]
16. Joshi, K.D.; Kataoka, K. pSMART: A lightweight, privacy-aware service function chain orchestration in multi-domain NFV/SDN. *Comput. Netw.* **2020**, *178*, 107295. [CrossRef]
17. Paolucci, F.; Cugini, F.; Castoldi, P.; Osinski, T. Enhancing 5G SDN/NFV Edge with P4 Data Plane Programmability. *IEEE Netw.* **2021**, *35*, 154–160. [CrossRef]
18. Vijayakumar, P.; Bose, S.; Kannan, A. Chinese remainder Theorem based centralised group key management for secure multicast communication. *IET Inf. Secur.* **2014**, *8*, 179–187. [CrossRef]
19. Sherman, A.T.; McGrew, D.A. Key establishment in large dynamic groups using one-way function trees. *IEEE Trans. Softw. Eng.* **2003**, *29*, 444–458. [CrossRef]
20. Ezekiel, S.; Divakaran, D.M.; Gurusamy, M. Dynamic attack mitigation using SDN. In Proceedings of the 2017 27th International Telecommunication Networks and Applications Conference (ITNAC), Melbourne, VIC, Australia, 22–24 November 2017; pp. 1–6. [CrossRef]
21. Babiceanu, R.F.; Seker, R. Cyber resilience protection for industrial internet of things: A software-defined networking approach. *Comput. Ind.* **2019**, *104*, 47–58. [CrossRef]
22. Babiceanu, R.F.; Seker, R. Cybersecurity and resilience modelling for software-defined networks-based manufacturing applications. In *Service Orientation in Holonic and Multi-Agent Manufacturing*; Springer: Berlin/Heidelberg, Germany, 2017; Volume 694, pp. 167–176. [CrossRef]
23. Piedrahita, A.F.M.; Gaur, V.; Giraldo, J.; Cardenas, A.A.; Rueda, S.J. Leveraging Software-Defined Networking for Incident Response in Industrial Control Systems. *IEEE Softw.* **2017**, *35*, 44–50. [CrossRef]
24. Madhawa, S.; Balakrishnan, P.; Arumugam, U. Employing invariants for anomaly detection in software defined networking based industrial internet of things. *J. Intell. Fuzzy Syst.* **2018**, *35*, 1267–1279. [CrossRef]
25. Mansour, A.; Azab, M.; Rizk, M.R.M.; Abdelazim, M. Biologically-inspired SDN-based Intrusion Detection and Prevention Mechanism for Heterogeneous IoT Networks. In Proceedings of the 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 1–3 November 2018; pp. 1120–1125. [CrossRef]
26. Chung, J.; Jung, E.-S.; Kettimuthu, R.; Rao, N.S.; Foster, I.T.; Clark, R.; Owen, H. Advance reservation access control using software-defined networking and tokens. *Future Gener. Comput. Syst.* **2018**, *79*, 225–234. [CrossRef]
27. Sharma, P.K.; Park, J.H.; Jeong, Y.-S.; Park, J.H. SHSec: SDN based Secure Smart Home Network Architecture for Internet of Things. *Mob. Networks Appl.* **2019**, *24*, 913–924. [CrossRef]
28. Demetriou, S.; Zhang, N.; Lee, Y.; Wang, X.; Gunter, C.A.; Zhou, X.; Grace, M.C. HanGuard: SDN-Driven Protection of Smart Home WiFi Devices from Malicious Mobile Apps. In Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, Boston, MA, USA, 18–20 July 2017; Volume 2017, pp. 122–133. [CrossRef]
29. Caprolu, M.; Raponi, S.; Di Pietro, R. FORTRESS: An efficient and distributed firewall for stateful data plane SDN. *Secur. Commun. Netw.* **2019**, *2019*, 6874592. [CrossRef]
30. Ge, M.; Hong, J.B.; Yusuf, S.E.; Kim, D.S. Proactive defense mechanisms for the software-defined Internet of Things with non-patchable vulnerabilities. *Future Gener. Comput. Syst.* **2018**, *78*, 568–582. [CrossRef]
31. Zarca, A.M.; Bernabe, J.B.; Skarmeta, A.; Calero, J.M.A. Virtual IoT HoneyNets to mitigate cyberattacks in SDN/NFV-Enabled IoT networks. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 1262–1277. [CrossRef]
32. Wang, S.; Gomez, K.; Sithamparanathan, K.; Asghar, M.R.; Russello, G.; Zanna, P. Mitigating ddos attacks in sdn-based iot networks leveraging secure control and data plane algorithm. *Appl. Sci.* **2021**, *11*, 929. [CrossRef]
33. Wani, A.; Revathi, S.; Khaliq, R. SDN-based intrusion detection system for IoT using deep learning classifier (IDSIoT-SDL). *CAAI Trans. Intell. Technol.* **2021**, *6*, 281–290. [CrossRef]
34. Aslam, M.; Ye, D.; Tariq, A.; Asad, M.; Hanif, M.; Ndzi, D.; Chelloug, S.A.; Elaziz, M.A.; Al-Qaness, M.A.A.; Jilani, S.F. Adaptive Machine Learning Based Distributed Denial-of-Services Attacks Detection and Mitigation System for SDN-Enabled IoT. *Sensors* **2022**, *22*, 2697. [CrossRef]

35. Burmester, M.; Desmedt, Y. A secure and efficient conference key distribution system. *Lect. Notes Comput. Sci.* **1995**, *950*, 275–286. [CrossRef]
36. Waller, D.; Harder, E.; Agee, R. Key Management for Multicast: Issues and Architectures. 1999. Available online: https://www.rfc-editor.org/rfc/rfc2627 (accessed on 9 March 2023).
37. Ku, W.C.; Chen, S.M. An improved key management scheme for large dynamic groups using one-way function trees. In Proceedings of the 2003 International Conference on Parallel Processing Workshops, Kaohsiung, Taiwan, 6–9 October 2003; Volume 2003, pp. 391–396. [CrossRef]
38. Xu, X.; Wang, L.; Youssef, A.; Zhu, B. Preventing collusion attacks on the one-way function tree (OFT) scheme. In *Applied Cryptography and Network Security*; Springer: Berlin/Heidelberg, Germany, 2007; Volume 4521, pp. 177–193. [CrossRef]
39. Sun, Y.; Chen, M.; Bacchus, A.; Lin, X. Towards collusion-attack-resilient group key management using one-way function tree. *Comput. Netw.* **2016**, *104*, 16–26. [CrossRef]
40. Festijo, E.; Jung, Y.; Peradilla, M. Software-defined security controller-based group management and end-to-end security management. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 3365–3382. [CrossRef]
41. Mansour, A.; Malik, K.M.; Alkaff, A.; Kanaan, H. ALMS: Asymmetric Lightweight Centralized Group Key Management Protocol for VANETs. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 1663–1678. [CrossRef]
42. Tiloca, M.; Dini, G.; Rizki, K.; Raza, S. *Group Rekeying Based on Member Join History*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 19, pp. 343–381. [CrossRef]
43. Zhang, S.; Han, S.; Zheng, B.; Han, K.; Pang, E. Group Key Management Protocol for File Sharing on Cloud Storage. *IEEE Access* **2020**, *8*, 123614–123622. [CrossRef]
44. Tamizhselvan, C. A novel communication-aware adaptive key management approach for ensuring security in IoT networks. *Trans. Emerg. Telecommun. Technol.* **2022**, *2022*, e4605. [CrossRef]
45. Komninos, N.; Philippou, E.; Pitsillides, A. Survey in smart grid and smart home security: Issues, challenges and countermeasures. *IEEE Commun. Surv. Tutor.* **2014**, *16*, 1933–1954. [CrossRef]
46. Mantas, G.; Lymberopoulos, D.; Komninos, N. Security in Smart Home Environment. In *Wireless Technologies for Ambient Assisted Living and Healthcare: Systems and Applications*; IGI Global: Hershey, PA, USA, 2010; pp. 170–191. [CrossRef]
47. Chen, Y.-W.; Wang, J.-T.; Chi, K.-H.; Tseng, C.-C. Group-based authentication and key agreement. *Wirel. Pers. Commun.* **2012**, *62*, 965–979. [CrossRef]
48. Ouaissa, M.; Rhattoy, A.; Lahmer, M. Group access authentication of machine to machine communications in LTE networks. In Proceedings of the Second International Conference on Internet of things, Data and Cloud Computing, Cambridge, UK, 22–23 March 2017. [CrossRef]
49. Mininet for SDN. Available online: https://github.com/mininet/mininet (accessed on 9 March 2023).