sensors

Article

Integrated Management of Network Address Translation, Mobility and Security on the Blockchain Control Plane

Younchan Jung * and **Ronnel Agulto**

School of Information, Communications and Electronics Engineering, The Catholic University of Korea, 43 Jibong-ro, Bucheon-si, Gyeonggi-do 14662, Korea; ronnelagulto@catholic.ac.kr

* Correspondence: ycjung@catholic.ac.kr; Tel.: +82-2-2164-4364

Received: 5 December 2019; Accepted: 19 December 2019; Published: 21 December 2019

Abstract: Currently, the dual use of IPv4 and IPv6 is becoming a problem. In particular, Network Address Translation (NAT) is an important issue to be solved because of traversal problems in end-to-end applications for lots of mobile IoT devices connected to different private networks. The vertical model is typically used to solve NAT, mobility and security issues for them. However, the existing vertical model has limitations because it handles NAT, mobility and security management one by one. This paper proposes a Blockchain-based Integrated Network Function Management (BINFM) scheme where the NAT, mobility, and security management are handled at once. The proposed scheme is advantageous in that by using blockchain and the Query/Reply mechanism, each peer can easily obtain the necessary parameters required to handle the NAT, mobility, and security management in a batch. In addition, this paper explains how our proposed scheme guarantees secure end-to-end data transfers with the use of one time session key. Finally, it is proved that the proposed scheme improves performance on latency from the viewpoints of mobility and security compared to the existing vertical model.

Keywords: NAT management; smart mobility; security management; blockchain-based management; integrated management

1. Introduction

One of the main reasons to extend the IP address space in IPv6 is to give Internet of Things (IoT) devices a platform to operate on for solving scalability issues [1–3]. The purported 400% increase in growth in the last five years sheds some light on how much exponential IoT growth we can expect to see in the next several decades [4]. However, its slow replacement of IPv4 can cause a problem especially for the demand to construct smart cities in a decade [5]. Recent smart cities need to allow IPv4 network-connected IoT devices to connect each other. As of now, Network Address Translation (NAT) solves connectivity issues for the IPv4 network-connected IoT devices [6–8]. NAT has one accessible public address which will be shared among End Nodes (ENs) inside the private network. NAT essentially extends internal addressing from the global IP addressing used over the Internet. NAT provides network resources to get over a shortage of the address space by mapping relatively public IP addresses to private IP addresses. However, the non-standardized characteristics of NAT cause traversal problems especially with the development of peer-to-peer applications for small IoT devices [9].

Considering that IPv6 allows IPv6 network-connected IoT devices to be uniquely addressable, despite the use of NAT and IPv4 private IP addresses, the mobile IoT devices, which enter private networks, must be allowed to solve a connectivity issue. Then, this enables increasing mobility for mobile IoT devices connected to private networks in the smart city on the condition that a smart mobility management is provided [10–12].

In recent decades, preserving privacy and ensuring the security of data have emerged as important issues as confidential information or private data may be revealed by powerful data mining tools [13–15]. Therefore, if hackers attack a smart city with lots of IoT devices, the outcome could be far more catastrophic. It is argued that IPv6 offers better security solutions than IPv4, largely due to IPSec, with which IPv6 operates. It is known that widespread adoption of IPv6 will make man-in-the-middle attacks significantly more difficult [16]. IPSec, which works on a layer 3 plane, i.e., the network layer, aims to provide application-layer security in batch by means of securing IP-layer. However, it poses a difficult problem to run end-to-end encryption because of its difficult key exchange protocol between end-to-end IoT peers [17]. Small IoT end points will face a burden when they handles security association data to secure the layer 3 datagram services. So this paper is based on the idea that IPSec is difficult to be realized for mobile IoT devices with the private IP addresses.

As depicted in Figure 1, the existing vertical model starts with the NAT management followed by mobility management. Once the NAT and mobility management are made, the security management procedure begins.

Static assignment of IP addresses gives adversaries significant advantage to remotely scan networks and identify their targets accurately and quickly. As traditional approaches against this attack, the IP address assignment scheme based on DHCP or NAT has been used. However, they are insufficient to provide proactive countermeasures because the IP mutation is infrequent and traceable. Recently, OpenFlow Random Host Mutation (OFRHM), in which the OpenFlow controller frequently assigns each host a random virtual IP that is translated to/from the real IP of the host, has been proposed [18]. The real IP remains untouched, so IP mutation is completely transparent to end-hosts. Implementation of this technique requires two major components: (i) subnet gateways to perform [real IP]/[virtual IP] translation, and (ii) a central management authority that coordinates mutation across the network. Software-defined networking (SDN) provides a flexible infrastructure for developing and managing random host mutation efficiently and with minimal operational overhead. Mobile Edge Computing (MEC) has emerged as a vital solution to offer computing resources at the edge of the network and in close vicinity to the mobile end-users [19]. The concept of MEC was motivated by the unprecedented growth of mobile traffic. Also, as another 5G enabling technology, SDN is complementing the MEC advancement. The MEC environment can substantially benefit from SDN technology. The routing of the ENs' offloading traffic can be performed in the control plane, which is implemented within the SDN controller. The commonality of OFRHM and MEC is to use SDN as a means and the difference of them is related to the different goals of ENs' frequent IP mutations and offloading traffic. Differently from SDN-based approaches, blockchain-based decentralized management schemes have been proposed [20,21]. In this paper, we also use the blockchain technology instead of SDN. Also, the goal of this paper is to make the integrated management of NAT, mobility, and security perform efficiently.

In this paper, three issues are addressed for small IoT devices which enter the private networks. First, a smart NAT management is needed in order to manage the private addressing of the local ENs in the private region and solve the NAT traversal issues. Second, the issue of mobility management, which focuses on ENs that use private IP addresses, should be solved. Most of the existing mobility management schemes only deal with the tracking of the location of the EN but not its transport addresses, i.e., the internal private address and mapped NAT address. Therefore, NAT management needs to collaborate with mobility management. Third, this paper solves security issue without the use of IPSec. During a session between two peers, they can use one time session key, which is delivered by the key exchange procedure with the blockchain's help. Differently from the vertical model, this paper aims to use a Blockchain-based Integrated Network Function Management (BINFM) system where the blockchain control plane functions as a platform to deal with the NAT, mobility and security management in batch. In the BINFM system, the session initiator can obtain the transport address-related and security-related information of the session responder just at the time when the initiator calls the responder. Jung et al. proposed a blockchain-based security management scheme to make a real-time packet key exchange perform better [22]. As a similar approach, for a certain end node, its security-related information are already stored and updated in the blockchain. Therefore, any end node, which wants to establish a new session to its peer, can obtain security parameters to derive the one time session key with the help of the blockchain while the vertical model requires the extra hand shaking procedure of key agreement in order to complete the security management.

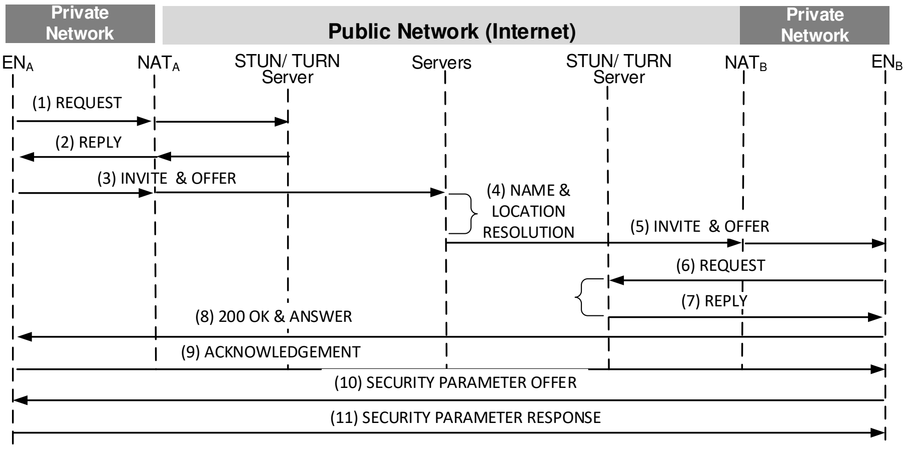


Figure 1. Vertical model for NAT, Mobility and Security management.

The BINFM scheme uses one of the most innovative features of the blockchain, in which there is no central server running. It operates through the network of blockchain control plane that the super nodes (SNs) constitute. Here, the BINFM scheme includes Query/Reply mechanism from the viewpoints of ENs. Using the Query/Reply mechanism, the EN obtains the transport address-related information, which solve the NAT and mobility management, and security-related information from its nearest SN. Therefore, this idea of Integrated Management (IM) gives significantly advantageous effects on NAT, mobility and security controls by reducing the system complexity and latency taken for mobility and security controls. Furthermore, the use of one time session key makes data flow over BINFM system more secure.

The rest of this paper is organized as follows. Section 2 proposes the blockchain-based architecture for smart NAT management. In Section 3, this paper explains how to process a transaction to create a block and query/reply mechanism needed to access the transaction information from the blockchain. Section 4 describes the improvement effects of the proposed management system. This paper concludes in Section 5.

2. Blockchain-Based Network Architecture for the Integrated Management

2.1. Proposed Network Architecture for Blockchain Control Plane

The blockchain keeps the database associating with the current transport address-related information for every mobile ENs from the NAT and mobility management viewpoints. For a specific application in the EN_A behind the private network, a set of transport address-related information is assigned, that is, [private IP address, private source port number, NATed public IP address, NATed port number] = [$IP\_EN_A\_Pri$, $Port\_EN_A\_Pri$, $IP\_EN_A\_NAT_{ED}$, $Port\_EN_A\_NAT_{ED}$]. For the corresponding EN_B behind the different private network, the transport address-related information of [$IP\_EN_B\_Pri$, $Port\_EN_B\_Pri$, $IP\_EN_B\_NAT_{ED}$, $Port\_EN_B\_NAT_{ED}$] are assigned. The role of the blockchain network is to enable both of session initiator and session responder to understand the transport address-related information for each other via the blockchain network. Query(to Blockchain)/Reply (from Blockchain) mechanism enables the EN to obtain information necessary for integrated management. In order to give an answer for this Query, the blockchain network requires to support the registration process that the EN's latest state information enter the blockchain. Figure 2 shows the BINFM network architecture, which can be explained as follows:

1. The EN is identified by the hash address derived from the public part of a public-private cryptographic key. The private part of the key is under the control of the EN.
2. The EN is responsible to update its Integrated Management (IM)-related information by pushing it into the blockchain. When an EN changes its private network, it sends a new registration transaction, that is, ToALL Tx, to the nearest super node (SN). After an SN receive the transaction message, it broadcasts the message to all SNs. Each transaction message contains several data fields for NAT, mobility and security management, which will be described in the next section.
3. The SN collects new ToALL Txs into a block and performs on solving the proof-of-work for its block. When an SN finds a proof-of-work for the ToALL Tx, it broadcasts the resultant block to all SNs. SNs imply their acceptance of the block by working on creating the next necessary block in the chain, using the hash of the accepted block as the previous hash. SNs will always keep working on extending it. The latency to extend a new block in the blockchain is closely related to the latency that takes the registration process to be completed. Therefore, this latency needs to be reduced as much as possible. Therefore, in this paper, it is assumed that next necessary block is created every 1 second on average and extended to the existing blockchain. This assumption is based on our experimental results using our blockchain testbed.
4. The EN uses the Query/Reply mechanism to obtain the peer's IM-related information from the blockchain. When an EN initiates to setup a session to its peer EN, it needs the peer EN's ToALL Tx information. To obtain this Tx information, it sends a query message to the nearest SN to gather the peer EN's transport address to reach there. Then, the SN searches the Tx Search Table (TST) and finds the corresponding transaction data from its blockchain. Then, it returns the requested transaction information to the EN.

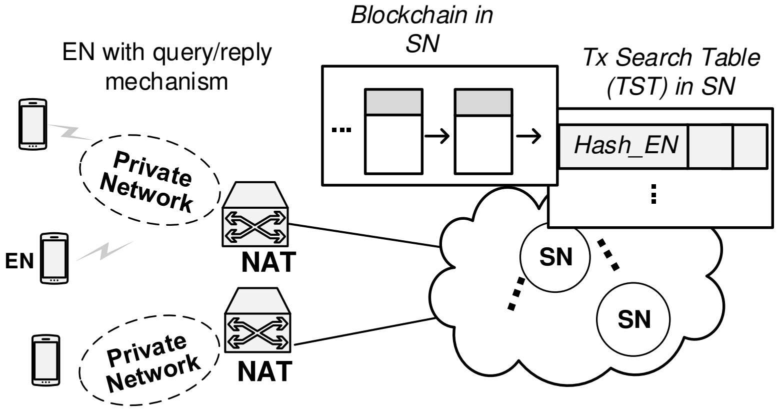


Figure 2. Proposed network architecture for Blockchain-based integrated management.

2.2. Requirements for the Proposed System

Private IP addresses must be configured automatically for new ENs that move from one network to another. Dynamic Host Configuration Protocol (DHCP) server maintains a pool of private IP addresses and leases an address to any DHCP-enabled EN when it starts up on the network. A

DHCP-enabled EN, upon accepting a lease offer, receives a valid private IP address for the private network to which it is currently connecting. There are additional parameters that a DHCP server is configured to assign to ENs. In the proposed BINFM scheme, the NAT address ($IP\_EN\_NAT_{ED}$, which is one accessible public address that will be shared among ENs inside the private network, is included in those parameters DHCP server offers. Figure 3 shows that the DHCP reply message contains the offered private address and the NAT address which will be used as the EN's source address when its packet enters the public network.

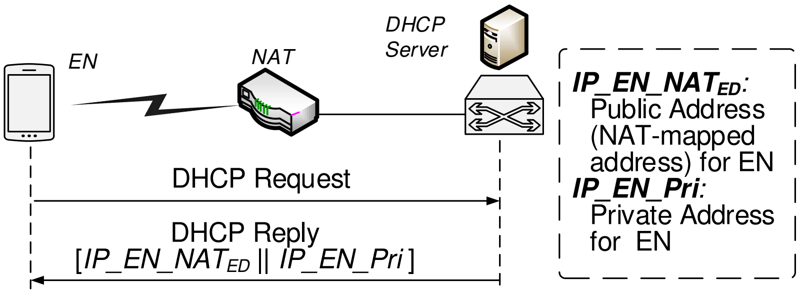

EN
NAT
DHCP Server
DHCP Request
DHCP Reply
[IP_EN_NAT_ED || IP_EN_Pri]
IP_EN_NAT_ED: Public Address (NAT-mapped address) for EN
IP_EN_Pri: Private Address for EN

Figure 3. Obtaining public NAT address during private address assignment stage.

When EN sends a packet using its source private IP address and port number (IP_EN_Pri: $Port\_EN\_Pri$) to destination IP address and port number (IP_Dest: $Port\_Dest$), the NAT creates a map for EN's private address and port number (IP_EN_Pri: $Port\_EN\_Pri$) by assigning public $IP\_EN\_NAT_{ED}$ and $Port\_EN\_NAT_{ED}$ as public address and port number, respectively. Therefore, incoming packets from [IP_Dest: $Port\_Dest$] destined to [$IP\_EN\_NAT_{ED}$: $Port\_EN\_NAT_{ED}$] are forwarded to [IP_EN_Pri: $Port\_EN\_Pri$]. As depicted in Figure 4, the BINFM scheme requires the important condition that $Port\_EN\_NAT_{ED}$ should be derived from the hash function of IP_EN_Pri and $Port\_EN\_Pri$. EN is aware that NAT devices use the NAT port assignment function of **H16** where the first 16 bits are taken from the hash value.

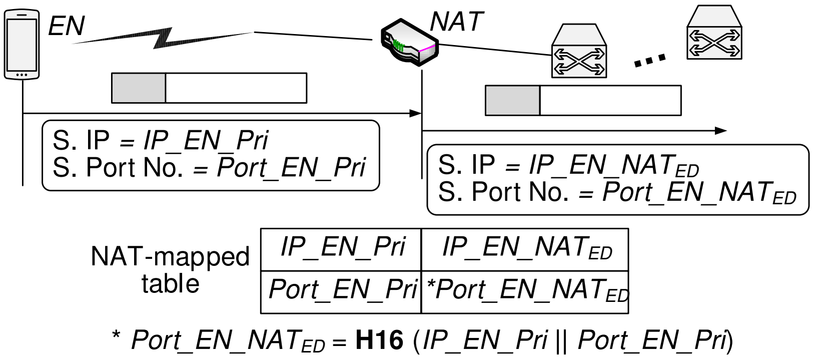
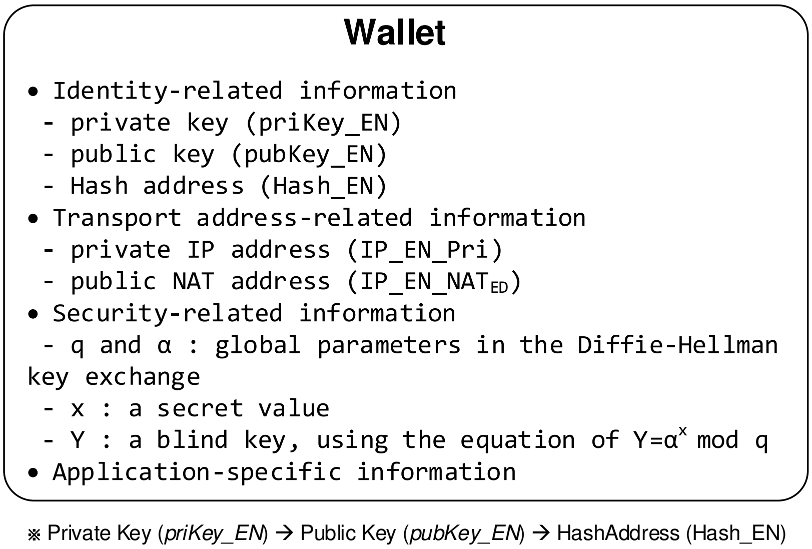


Figure 4. Public NAT port number determined as a function of EN's private IP address and port number.

3. Blockchain-Based NAT, Mobility and Security Management

3.1. Smart Wallets for the End Nodes

ENs such as small IoT devices have smart wallets. As shown in Figure 5, smart wallet contains its own identity-related, transport address-related and security-related information. So the EN is responsible for registering this information with the blockchain maintained in the SNs. To obtain this information for the other side from the blockchain, the EN uses Query/Reply mechanism. Therefore, each side can easily obtain the necessary parameters of the other side required to handle the NAT, mobility and security management to establish and maintain a secure session between two peers which entered two different private networks. Here, 'smart' wallet means that the EN takes advantage of the blockchain's merits without maintaining the blockchain data structure.



Wallet

- Identity-related information
  - private key (priKey_EN)
  - public key (pubKey_EN)
  - Hash address (Hash_EN)
- Transport address-related information
  - private IP address (IP_EN_Pri)
  - public NAT address ($\text{IP\_EN\_NAT}_{\text{ED}}$)
- Security-related information
  - q and α : global parameters in the Diffie-Hellman key exchange
  - x : a secret value
  - Y : a blind key, using the equation of $Y=\alpha^{x} \bmod q$
- Application-specific information

※ Private Key (*priKey_EN*) → Public Key (*pubKey_EN*) → HashAddress (Hash_EN)

Figure 5. Smart wallet information.

3.2. BINFM Transactions and Registration Procedure

Each EN's latest state information resides in its own wallet. however, all ENs' information are stored in a distributed database called the blockchain, which stores a secure list of all ToALL transactions sent by them. The BINFM transaction is defined as the EN's state record during the period of temporally assigned private IP address. Therefore, the transaction change rate is the same as the private IP address change rate. This means that EN sends its ToALL Tx to the network whenever it moves and obtains a new private IP address. When a handover occurs during a call session between two ENs, the EN, which changes its private IP address, also issues new ToALL Tx to the network. The transaction consists of Transaction Input (TxIn) and Transaction Output (TxOut). The TxIn contains the signature and the public key computed from the EN's private key which creates the transaction. The first field of TxOut contains the hash address that identifies the owner of this transaction. Figure 6 explains the ToALL Tx structure.

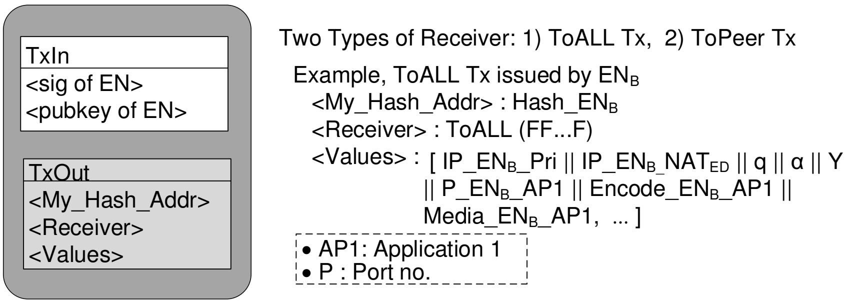

TxIn
<sig of EN>
<pubkey of EN>
TxOut
<My_Hash_Addr>
<Receiver>
<Values>
Two Types of Receiver: 1) ToALL Tx, 2) ToPeer Tx
Example, ToALL Tx issued by EN_B
<My_Hash_Addr> : Hash_EN_B
<Receiver> : ToALL (FF...F)
<Values> : [IP_EN_B_Pri || IP_EN_B_NAT_ED || q || α || Y || P_EN_B_AP1 || Encode_EN_B_AP1 || Media_EN_B_AP1, ...]
• AP1: Application 1
• P : Port no.

Figure 6. BINFM transaction structure.

Figure 7 shows the registration procedure of the IM information. Each EN updates their state information including the transport address information by sending ToALL Tx whenever it moves to the new private network. The first SN in the network that receives the Tx verifies the sent Tx if it is a valid Tx. If the Tx is correct, the SN relays it to other SNs in the network.

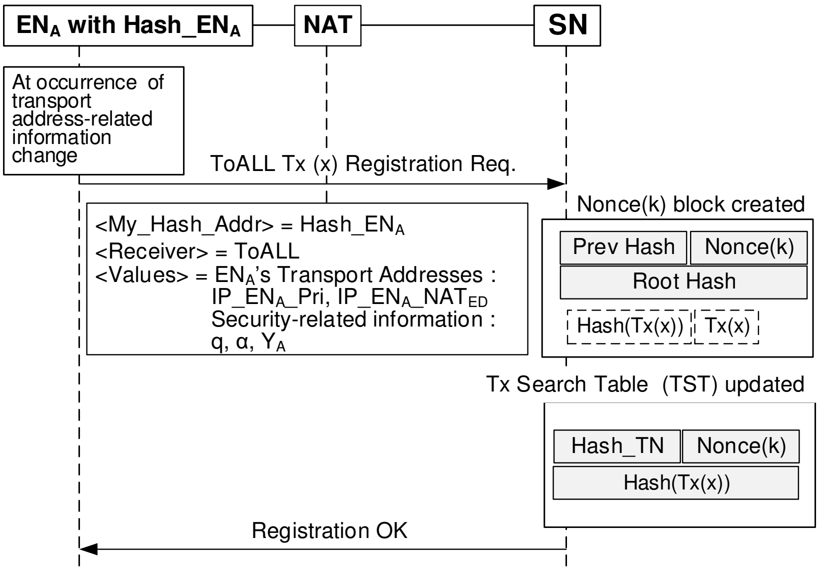

EN_A with Hash_EN_A
NAT
SN
At occurrence of transport address-related information change
ToALL Tx (x) Registration Req.
Nonce(k) block created
<My_Hash_Addr> = Hash_EN_A
<Receiver> = ToALL
<Values> = EN_A's Transport Addresses : IP_EN_A_Pri, IP_EN_A_NAT_ED
Security-related information : q, α, Y_A
Prev Hash
Nonce(k)
Root Hash
Hash(Tx(x))
Tx(x)
Tx Search Table (TST) updated
Hash_TN
Nonce(k)
Hash(Tx(x))
Registration OK

Figure 7. BINFM registration process.

3.3. Blockchain-Based Integrated Management Procedure

Figure 8 shows the proposed BINFM-based Integrated Management procedure to establish a secure session between two peers that stay in two different private networks. When EN_A with the $Hash\_EN_A$ wants to establish a session with EN_B with the $Hash\_EN_B$ (session responder), EN_A first uses the Query/Reply mechanism. EN_A sends (a) Query message which contains $Hash\_EN_B$ to the nearest SN. When an SN, which has the blockchain information, receives the Query, it seeks the corresponding Tx for $Hash\_EN_B$ with the help of TST. The SN sends back (b) Reply message containing the Tx information of EN_B, that is, global parameters of q and α and EN_B's blind key (Y_B) as well as the transport address-related information ($IP\_EN_B\_NAT_{ED}$ and $IP\_EN_B\_Pri$). Here, $Y_B = \alpha^{X_B} mod\ q$ where X_B is a secret value of EN_B. Now, EN_A can send (c) Session Request message to $IP\_EN_B\_Pri$ via $IP\_EN_B\_NAT_{ED}$. This message contains EN_A's hash address of $Hash\_EN_A$. When NAT_B receives the packet, it translates the destination IP address and destination port number of the datagram NAT_A sent, as $IP\_EN_B\_Pri$ and $Port\_EN_B\_Pri$. When EN_B receives the Session Request message, it extracts the

EN_A's hash address of $Hash\_EN_A$ from the message. Now, the EN_B sends (d) Query message which contains the $Hash\_EN_A$ to the nearest SN. When an SN receives the Query, it seeks the corresponding ToALL Tx published from the $Hash\_EN_A$. The SN sends back (e) Reply message containing the Tx information for EN_A. Then, EN_A obtains global parameters of q and α and EN_A's blind key (Y_A) as well as the physical addresses of $IP\_EN_A\_NAT_{ED}$ and $IP\_EN_A\_Pri$. Here, $Y_A = \alpha^{X_A} mod\ q$ where X_A is a secret value of EN_A. Now, EN_B is ready to send its datagrams to $IP\_EN_A\_Pri$ via $IP\_EN_A\_NAT_{ED}$. When NAT_A receives those datagrams, it translates the destination IP address and destination port number as $IP\_EN_A\_Pri$ and $Port\_EN_A\_Pri$.

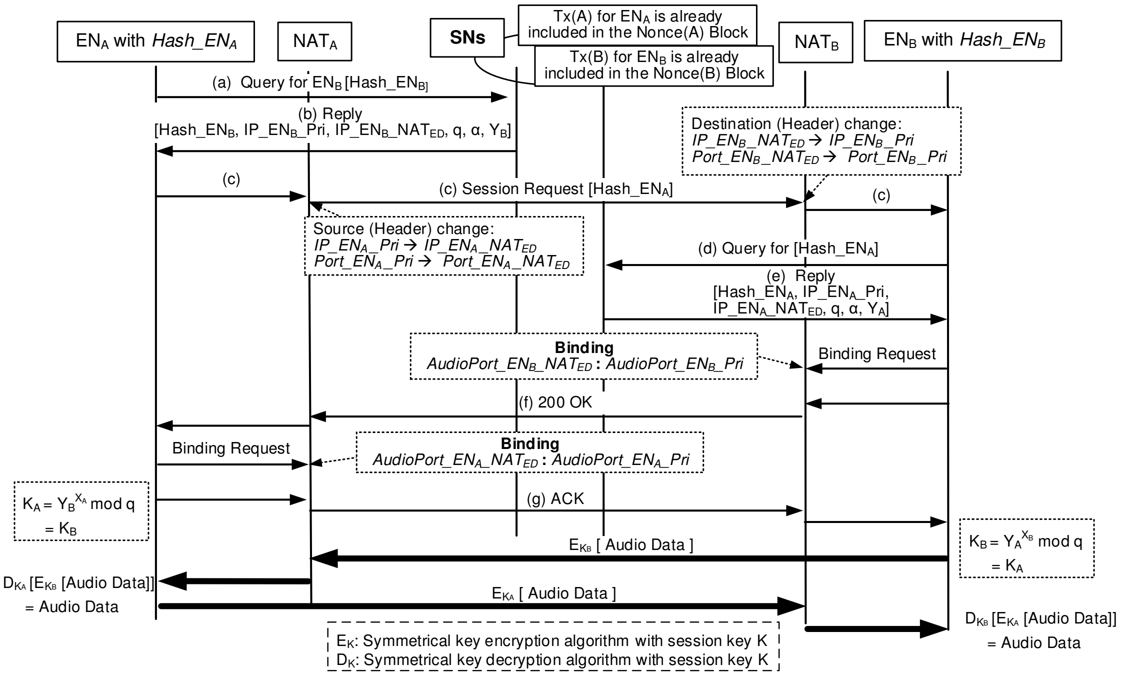

EN_A with Hash_EN_A
NAT_A
SNs
Tx(A) for EN_A is already included in the Nonce(A) Block
Tx(B) for EN_B is already included in the Nonce(B) Block
NAT_B
EN_B with Hash_EN_B
(a) Query for EN_B [Hash_EN_B]
(b) Reply [Hash_EN_B, IP_EN_B_Pri, IP_EN_B_NAT_ED, q, α, Y_B]
Destination (Header) change: IP_EN_B_NAT_ED → IP_EN_B_Pri Port_EN_B_NAT_ED → Port_EN_B_Pri
(c)
(c) Session Request [Hash_EN_A]
(c)
Source (Header) change: IP_EN_A_Pri → IP_EN_A_NAT_ED Port_EN_A_Pri → Port_EN_A_NAT_ED
(d) Query for [Hash_EN_A]
(e) Reply [Hash_EN_A, IP_EN_A_Pri, IP_EN_A_NAT_ED, q, α, Y_A]
Binding AudioPort_EN_B_NAT_ED : AudioPort_EN_B_Pri
Binding Request
(f) 200 OK
Binding Request
Binding AudioPort_EN_A_NAT_ED : AudioPort_EN_A_Pri
K_A = Y_B^X_A mod q = K_B
(g) ACK
E_KB [Audio Data]
K_B = Y_A^X_B mod q = K_A
D_KA [E_KB [Audio Data]] = Audio Data
E_KA [Audio Data]
D_KB [E_KA [Audio Data]] = Audio Data
E_K: Symmetrical key encryption algorithm with session key K
D_K: Symmetrical key decryption algorithm with session key K

Figure 8. Blockchain-based NAT, mobility and security management procedure.

From security management viewpoints, EN_A and EN_B maintain X_A and X_B, respectively. After each Query/Reply procedure, EN_A and EN_B are ready to use Y_B and Y_A, respectively. Then, EN_A computes the one time session key of K_A using the equation of ${Y_B}^{X_A}$ $mod\ q$ while EN_B computes the one time session key of K_B using the equation of ${Y_A}^{X_B}$ $mod\ q$. Here, K_A is equal to K_B. Now, EN_A can encrypt its datagrams using the session key which results in E_{K_A}[Audio Data] where E_{K_A} is any symmetrical key encryption algorithm with the key K_A. Therefore, EN_A sends the encrypted datagrams to EN_B. When EN_B receives the encrypted datagrams from EN_A, it can decrypt those datagrams using the session key K_B which results in $D_{K_B}[E_{K_A}$[Audio Data]] = [Audio Data] where D_{K_B} is any symmetrical key decryption algorithm with the key K_B. As a result, bidirectional session traffic travel over the established secure sessions. Therefore, our blockchain-based scheme easily solves the problem of handling complex issues of NAT, mobility and security management. This advantage results from the fact that each peer can obtain the necessary parameters for peer-to-peer session establishment via a simple Query/Reply mechanism between an EN and its nearest SN.

3.4. Blockchain-Based Mid-Call Mobility Management Procedure

The mid-call mobility management is needed when either EN_A or EN_B changes its local private network during an on-going session. Figure 9 shows the mid-call mobility management operation for the case that EN_A changes its network during the on-going session with EN_B. When EN_A confronts with IP handover, it first sends IP handover Request message to the EN_B. This massage contains a new transport address-related information, that is, $[IP'\_EN_A\_NAT_{ED}$ and $IP'\_EN_A\_Pri]$. Next, a new ToALL Tx registration procedure starts to update EN_A's state information on the blockchain. When EN_B receives the Request message, it immediately uses the updated transport address information for EN_A. Then, both can keep on going the existing bidirectional session.

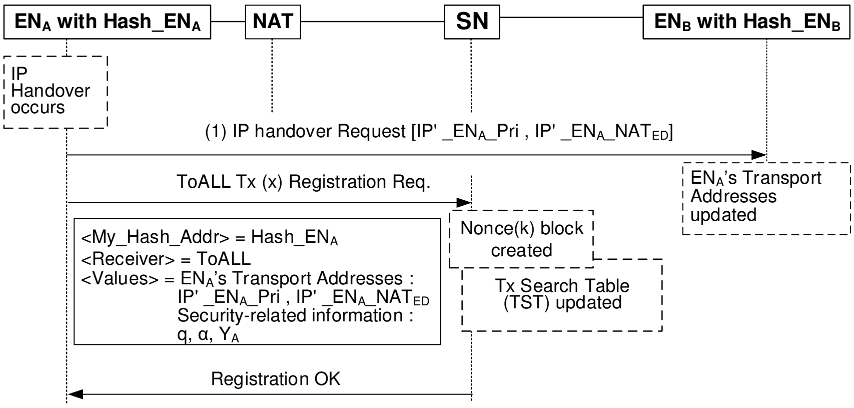

EN_A with Hash_EN_A
NAT
SN
EN_B with Hash_EN_B
IP Handover occurs
(1) IP handover Request [IP' _EN_A_Pri , IP' _EN_A_NAT_ED]
EN_A's Transport Addresses updated
ToALL Tx (x) Registration Req.
<My_Hash_Addr> = Hash_EN_A
<Receiver> = ToALL
<Values> = EN_A's Transport Addresses : IP' _EN_A_Pri , IP' _EN_A_NAT_ED
Security-related information : q, α, Y_A
Nonce(k) block created
Tx Search Table (TST) updated
Registration OK

Figure 9. Blockchain-based mid-call mobility management procedure.

3.5. Key Renewal Process During a Session

From security management viewpoints, any side can initiate to change the one time session key even during an on-going session. Figure 10 shows the key change operation for the case that EN_A needs to change its one time session key during the on-going session with EN_B. As a key change initiator, EN_A generates new secret value X'_A and computes $Y'_A = \alpha^{X'_A} mod\ q$. Also, EN_A prepares new one time session key $K'_A = {Y_B}^{X'_A} mod\ q$. Then, EN_A sends the Key Renewal Request message, which contains the blind key Y'_A, to EN_B. Once EN_B receives Y'_A, it computes the one time session key of K'_B using the equation of ${Y'_A}^{X_B}\ mod\ q$.

Now, EN_A can encrypt its datagrams using the new session key which results in $E_{K'_A}$[Audio Data]. Therefore, EN_A sends the encrypted datagrams to EN_B. When EN_B receives the encrypted datagrams from EN_A, it can decrypt those datagrams using the new session key K'_B which results in $D_{K'_B}[E_{K'_A}$[Audio Data]] = [Audio Data].

Because EN_A changes its secret value, it needs to update its security-related information in the blockchain. EN_A updates its state information including the security-related information by sending ToALL Tx to the SN.

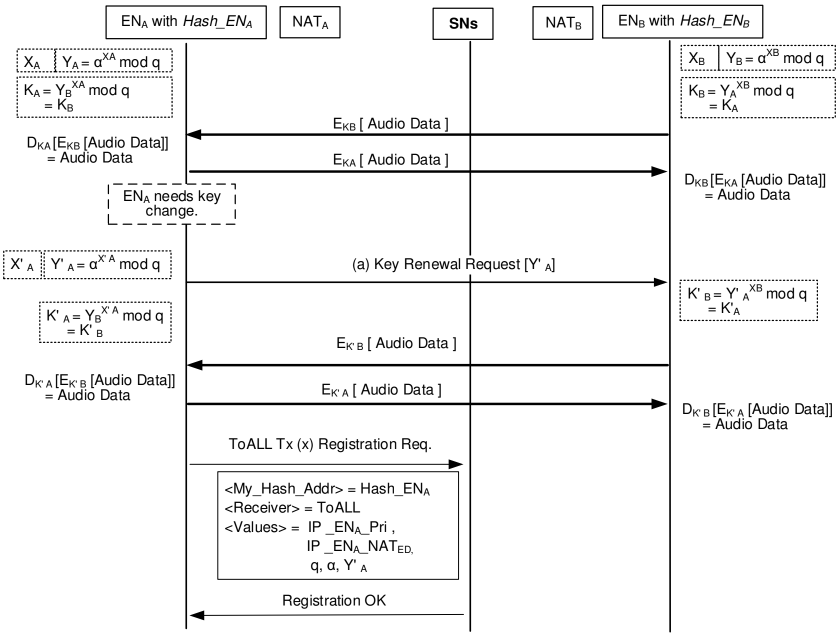

EN_A with $Hash\_EN_A$
NAT_A
SNs
NAT_B
EN_B with $Hash\_EN_B$
X_A | $Y_A = \alpha^{XA}$ mod q
$K_A = Y_B^{XA}$ mod q = K_B
X_B | $Y_B = \alpha^{XB}$ mod q
$K_B = Y_A^{XB}$ mod q = K_A
E_{KB} [Audio Data]
$D_{KA}[E_{KB}$ [Audio Data]] = Audio Data
E_{KA} [Audio Data]
$D_{KB}[E_{KA}$ [Audio Data]] = Audio Data
EN_A needs key change.
X'_A | $Y'_A = \alpha^{X'A}$ mod q
(a) Key Renewal Request [Y'_A]
$K'_B = Y'_A{}^{XB}$ mod q = K'_A
$K'_A = Y_B^{X'A}$ mod q = K'_B
$E_{K'B}$ [Audio Data]
$D_{K'A}[E_{K'B}$ [Audio Data]] = Audio Data
$E_{K'A}$ [Audio Data]
$D_{K'B}[E_{K'A}$ [Audio Data]] = Audio Data
ToALL Tx (x) Registration Req.
<My_Hash_Addr> = Hash_EN_A
<Receiver> = ToALL
<Values> = IP _EN_A_Pri , IP _EN_A_$NAT_{ED,}$ q, α, Y'_A
Registration OK

Figure 10. Key change event during a session.

4. Improvement Effects of Blockchain-Based Approaches

4.1. Comparisons between the Existing Vertical Model and the Proposed BINFM Model for the Pre-Call Mobility and Handover Management

Figure 1 shows a series of steps, which correspond to the pre-call mobility management procedure in the vertical model, to complete a secure session set up between two ENs where they are located within the different private networks. Here, each EN changes its location dynamically. Figure 11 shows a series of steps to handle a mid-call mobility management between two ENs at the circumstance that one of them changes its private IP address.

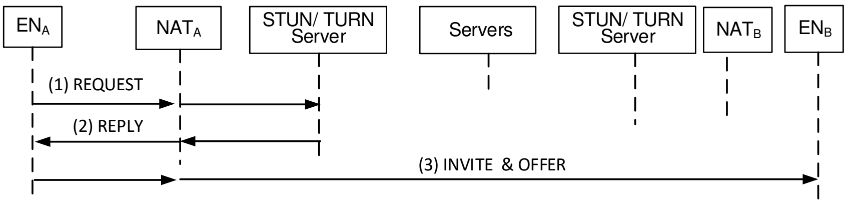


Figure 11. Mid-call mobility management procedure in the vertical model.

Figure 8 shows the proposed BINFM-based VoIP call setup procedure which corresponds to the pre-call mobility procedure in the proposed BINFM model. As shown in Figure 9, the mid-call mobility management in the proposed BINFM model is already described.

The following assumptions have been made to perform the comparative analysis with respect to total latency to complete the IM management. Three types of delay components exist, that is,

- T_I: intra-domain delay caused in intra-domain links,
- T_{II}: end-to-end delay caused in end-to-end path,
- T_{III}: delay caused to collaborate with the distributed servers, which are spread in inter-domain regions,

where T_{II} = $5T_I$ and T_{III} = $10T_I$. This assumption is based on the blockchain network architecture where the unit delay of T_I corresponds to the packet delay to travel from a certain EN to its nearest SN and the end-to-end path between two peers is longer by 5 times compared to the unit delay. Also, T_{III} is assumed to be twice compared with the end-to-end path delay because the delay of T_{III} includes delay components needed for searching processes in the distributed servers. Considering that with 4G networks, average latency is around 50 ms, the unit delay of T_I is set to 40 ms.

Table 1 compares the vertical model in Figure 1 with BINFM model in Figure 8. In the BINFM system, the Query/Reply procedures are only required to agree on necessary parameters to solve the issues relating to NAT, mobility and security management. As shown in Table 1, the pre-call mobility management latency requires 760 ms in the BINFM system compared to the vertical model, which needs the latency of 1440 ms for pre-call mobility management.

Table 1. Latency comparison for pre-call mobility management.

	BINFM Model (Figure 8)	**Vertical Model (Figure 1)**
Delay Components	T_I: (a), (b), (d), (e) T_{II}: (c), (f), (g) T_{III}: None	T_I: (1), (2), (3), (5), (6), (7) T_{II}:(8), (9), (10), (11) T_{III}: (4)
Latency	$4T_I + 3T_{II}$ (760 ms)	$6T_I + 4T_{II} + T_{III}$ (1440 ms)

As shown in Table 2, the vertical model yields a latency of 280 ms for mid-call mobility management. In the BINFM model, a mid-call mobility management needs the latency of 200 ms.

Table 2. Latency comparison for mid-call mobility management.

	BINFM Model (Figure 9)	**Vertical Model (Figure 11)**
Delay Components	T_I: None T_{II}: (1)	T_I: (1), (2) T_{II}: (3)
Latency	T_{II} (200 ms)	$2T_I + T_{II}$ (280 ms)

4.2. Comparisons between the Existing Vertical Model and the Proposed BINFM Model for the Security Management

As shown in Table 3, the BINFM model needs the latency of 200 ms to complete a new key agreement procedure between two peers during a session. the vertical model yields a latency of at least 400 ms for the same key management.

Table 3. Latency comparison for security management.

	BINFM Model (Figure 10)	**Vertical Model (Figure 1)**
Delay Components	T_{II}: (a)	T_{II}: (10), (11)
Latency	T_{II} (200 ms)	$2T_{II}$ (400 ms)

4.3. Comparisons between the Existing Vertical Model and the Proposed BINFM Model for the Signaling Overhead

This subsection analyzes the signaling overhead that is imposed in the overall system. Without loss of generosity, three types of signaling overhead can be assumed,

- S_I: intra-domain signaling overhead caused in intra-domain links,
- S_{II}: end-to-end signaling overhead caused in end-to-end path,
- S_{III}: signaling overhead caused to collaborate with the distributed servers, which are spread in inter-domain regions,

where S_{II} = $5S_I$ and S_{III} = $10S_I$. This condition is based on the same assumption to obtain the results shown in Table 1. The unit signaling overhead of S_I corresponds to the amount of signaling overhead for the Query message in Figure 8 to complete a mission. Inferring using the same method as Table 1, the overall signaling overhead requires $19S_I$ in the BINFM system compared to the vertical model, which needs the overall signaling overhead of $36S_I$ for completing the whole management to establish a secure session. It is found that the signaling overhead, which is imposed in the BINFM system, can be reduced to the level of 52% compared to the vertical model.

4.4. Complexity Analysis of the Proposed BINFM Model

If our BINFM approach can be implemented in real time or close to real time within a realistic networking environment, the complexity of the system can be explained in two ways. In the BINFM system, the role of the blockchain network is to enable two ENs as session initiator and session responder to agree on the mutual transport address-related and security-related information close to real-time. Therefore, blockchain information need to be updated as fast as possible when a certain EN issues a ToALL Tx. This latency is the same as the block creation period to extend a new block in the blockchain. Therefore, the latency to complete the registration process will be reduced as much as the block creation period decreases. In this paper, it is required to solve the complexity of the BINFM system in which the proof-of-work takes 1 second on average to succeed. Next, complexity is related to the Query/Reply mechanism. It starts to work by sending a Query Tx to the nearest SN. Then, the SN searches the corresponding transaction data from its blockchain with the Tx Search Table (TST)'s help and replies the searched transaction information. As the number of ENs increases and their movements increase, the complexity of finding information of the desired counterpart will increase. This complexity is closely related to the scalability of the system. It is beyond the scope of this paper.

5. Conclusions

Currently, the vertical model is typically used to solve Network Address Translation (NAT), mobility, and security issues for the mobile IoT devices where IPv4 and IPv6 are used together as a network layer protocol. However, the existing vertical model confronts with limitations in handling NAT, mobility and security management in batch. This paper proposed a Blockchain-based Integrated Management system where the the NAT and mobility management are handled together with the security management at once. This paper proved that our BINFM scheme is advantageous in terms of using the blockchain and Query/Reply mechanism, and each side can easily obtain the necessary parameters of the other side required to handle the NAT, mobility, and security management to establish and maintain a secure session between two peers which entered two different private networks. It was proved that the proposed scheme performs better from the viewpoints of pre-call mobility, mid-call mobility, pre-call security, and mid-call security control issues than the existing vertical model.

Author Contributions: Y.J. conceptualized the article, wrote the text and performed critical revision of the manuscript; R.A. provided support for interpretation of important content and for editing of the entire manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2017R1A2B4006086) and by the Research Fund, 2019 of the Catholic University of Korea.

Conflicts of Interest: The authors declare no conflict of interest.The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Savolainen, T.; Soininen, J.; Silverajan, B. IPv6 Addressing Strategies for IoT. *IEEE Sens. J.* **2013**, *13*, 3511–3519. [CrossRef]
2. Ziegler, S.; Crettaz, C.; Ladid, L.; Krco, S.; Pokric, B.; Skarmeta, A.F.; Jara, A.; Kastner, W.; Jung, M. IoT6 – Moving to an IPv6-Based Future IoT. In *The Future Internet*; Galis, A., Gavras, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 161–172.
3. Pongle, P.; Chavan, G. A survey: Attacks on RPL and 6LoWPAN in IoT. In Proceedings of the 2015 International Conference on Pervasive Computing (ICPC), Pune, India, 8–10 January 2015; pp. 1–6. [CrossRef]
4. Kaur, J.; Kaur, K. Internet of Things: A review on technologies, architecture, challenges, applications, future trends. *Int. J. Comput. Netw. Inf. Secur.* **2017**, *9*, 57. [CrossRef]
5. Mishra, S.; Singh, P.; Tiwari, A.K. Reliable Data Delivery with Extended IPV4 Using Low-Power Personal Area Network. In *Ambient Communications and Computer Systems*; Hu, Y.C., Tiwari, S., Mishra, K.K., Trivedi, M.C., Eds.; Springer: Singapore, 2019; pp. 25–35.
6. Kim, G.; Kim, J.; Lee, S. An SDN based fully distributed NAT traversal scheme for IoT global connectivity. In Proceedings of the 2015 International Conference on Information and Communication Technology Convergence (ICTC), Jeju Island, Korea, 28–30 October 2015; pp. 807–809. [CrossRef]
7. Patton, M.; Gross, E.; Chinn, R.; Forbis, S.; Walker, L.; Chen, H. Uninvited Connections: A Study of Vulnerable Devices on the Internet of Things (IoT). In Proceedings of the 2014 IEEE Joint Intelligence and Security Informatics Conference, The Hague, The Netherlands, 24–26 September 2014; pp. 232–235. [CrossRef]
8. Jung, Y.; Peradilla, M.; Saini, A. Software-defined Naming, Discovery and Session Control for IoT Devices and Smart Phones in the Constraint Networks.*Procedia Comput. Sci.* **2017**, *110*, 290–296. [CrossRef]
9. Wang, H.; Chen, C.; Lu, S. An SDN-based NAT Traversal Mechanism for End-to-end IoT Networking. In Proceedings of the 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS), Matsue, Japan, 18–20 September 2019; pp. 1–4. [CrossRef]
10. Bi, Y.; Han, G.; Lin, C.; Guizani, M.; Wang, X. Mobility Management for Intro/Inter Domain Handover in Software-Defined Networks. *IEEE J. Sel. Areas Commun.* **2019**, *37*, 1739–1754. [CrossRef]
11. Chai, H.S.; Choi, J.Y.; Jeong, J. An enhanced secure mobility management scheme for building iot applications. *Procedia Comput. Sci.* **2015**, *56*, 586–591. [CrossRef]
12. Fafolahan, E.M.O.; Pierre, S. A Seamless Mobility Management Protocol in 5G Locator Identificator Split Dense Small Cells. Available online: https://ieeexplore.ieee.org/abstract/document/8706642 (accessed on 20 December 2019).
13. Lin, C.W.; Fournier Viger, P.; Wu, L.; Gan, V.W.; Djenouri, Y.; Zhang, J. PPSF: An Open-Source Privacy-Preserving and Security Mining Framework. In Proceedings of the 2018 IEEE International Conference on Data Mining Workshops (ICDMW), Singapore, 17–20 November 2018; pp. 1459–1463. [CrossRef]
14. Lin, J.C.; Wu, J.M.; Fournier-Viger, P.; Djenouri, Y.; Chen, C.; Zhang, Y. A Sanitization Approach to Secure Shared Data in an IoT Environment. *IEEE Access* **2019**, *7*, 25359–25368. [CrossRef]
15. Fan, L.; Bonomi, L. Time Series Sanitization with Metric-Based Privacy. In Proceedings of the 2018 IEEE International Congress on Big Data (BigData Congress), San Francisco, CA, USA, 2–7 July 2018; pp. 264–267. [CrossRef]
16. Mohamad Noor, M.B.; Hassan, W.H. Current research on Internet of Things (IoT) security: A survey. *Comput. Netw.* **2019**, *148*, 283–294. [CrossRef]

17. Raza, S.; Voigt, T.; Jutvik, V. Lightweight IKEv2: a key management solution for both the compressed IPsec and the IEEE 802.15. 4 security. In Proceedings of the IETF Workshop on Smart Object Security, Paris, France, 23 March 2012; Volume 23.
18. Jafarian, J.H.; Al-Shaer, E.; Duan, Q. Openflow random host mutation: transparent moving target defense using software defined networking. In Proceedings of the First Workshop on Hot Topics in Software Defined Networks, Helsinki, Finland, 13 August 2012; pp. 127–132.
19. Mitsis, G.; Apostolopoulos, P.A.; Tsiropoulou, E.E.; Papavassiliou, S. Intelligent Dynamic Data Offloading in a Competitive Mobile Edge Computing Market. *Future Internet* **2019**, *11*, 118. [CrossRef]
20. Pop, C.; Cioara, T.; Antal, M.; Anghel, I.; Salomie, I.; Bertoncini, M. Blockchain Based Decentralized Management of Demand Response Programs in Smart Energy Grids. *Sensors* **2018**, *18*, 162. [CrossRef] [PubMed]
21. Dwivedi, A.; Srivastava, G.; Dhar, S.; Singh, R. A Decentralized Privacy-Preserving Healthcare Blockchain for IoT. *Sensors* **2019**, *19*, 326. [CrossRef] [PubMed]
22. Jung, Y.; Peradilla, M.; Agulto, R. Packet Key-Based End-to-End Security Management on a Blockchain Control Plane. *Sensors* **2019**, *19*, 2310. [CrossRef] [PubMed]