Secure and Energy-Efficient Handover in Fog Networks Using Blockchain-Based DMM

Vishal Sharma, Ilsun You, Francesco Palmieri, Dushantha Nalin K. Jayakody, and Jun Li

DMM depends on external mechanisms for handover security and uses a centralized device, which has obvious security and performance implications in flat architectures where hierarchical dependencies can introduce problems. We propose a new DMM schema based on the blockchain, capable of resolving hierarchical security issues without affecting the network layout, and also satisfying fully distributed security requirements with less consumption of energy.

Abstract

Modern fog network architectures, empowered by IoT applications and 5G communications technologies, are characterized by the presence of a huge number of mobile nodes, which undergo frequent handovers, introducing a significant load on the involved network entities. Considering the distributed and flat nature of these architectures, DMM can be the only viable option for efficiently managing handovers in these scenarios. The existing DMM solutions are capable of providing smooth handovers, but lack robustness from the security point of view. Indeed, DMM depends on external mechanisms for handover security and uses a centralized device, which has obvious security and performance implications in flat architectures where hierarchical dependencies can introduce problems. We propose a new DMM schema based on the blockchain, capable of resolving hierarchical security issues without affecting the network layout, and also satisfying fully distributed security requirements with less consumption of energy.

Fog Networks Enabled by IoT with Secure and Energy-Efficient Handover

The progressive miniaturization of hardware equipment and the availability of energy-efficient fifth generation (5G) wireless communication technologies are now enabling the massive deployment of new smart mobile devices (sensors, remote controllers, and actuators). This helps to realize ambient intelligence, assisted driving, e-health services, and other advanced applications, which will result in thousands of millions of mobile networked objects, continuously producing and consuming data, deployed across large geographic areas, and mutually interconnected through the ubiquitous Internet of Things (IoT). Such a huge scale introduces several challenges affecting traditional network-centric architectures, ranging from the variable demand for processing and storage resources, exceeding the capability of centralized, statically dimensioned solutions, to the impossibility of collecting the large volumes of traffic generated by millions of devices in a central location for timely processing and analysis. Fog-based architectures have been conceived to cope with these challenges by pushing intelligence, processing power, and communication capabilities down within the base stations' (BSs') local area networks, and hence as close as possible to where data are originated. This implies flattening the network to achieve better services and quality of experience by shifting many data collection and processing activities within fog service nodes or IoT gateways that, being located near data sources, end users, and remote control devices, allow for highly scalable low-latency services.

The resulting fog network architecture is sketched in Fig. 1, showing a BS that acts as a pivot for connecting mobile users and IoT devices to the public cloud or to multiple private clouds through specific access points (AP). Mobile IoT devices or user land terminals, here referred to as mobile nodes (MNs), can directly connect to APs (small cells) or, when using the 5G millimeter-wave (mmWave) band, join the network by associating with picocells (PCs) and femtocells (FCs). They can communicate with static IoT devices or service nodes, referred to as corresponding nodes (CNs), connected to the network through specific home gateways (HGWs). A fog server (FS) controls multiple fog nodes (FNs), allowing users to interact with the fog storage and runtime resources through specific communication protocols depending on network configuration, users' density, and the applications involved.

While such a flattened architecture enables extremely scalable high-speed services, it still presents several open security issues at the network level. Centralized authentication devices are needed for managing the security between the involved entities and ensuring trust in service exchanges. This results in a security architecture that is a hybrid between flattened services and hierarchical security. Spectrum widening and high transmission rates are helpful in overcoming latency issues in hierarchical authentication, but pave the way to various kinds of cyber attacks. Furthermore, since most of the mobile devices involved are typically constrained by limited battery power, energy efficiency in mobility management, immediately achievable through a reduction of the number of messages exchanged by the different entities involved, is also a fundamental concern.

MNs are typically involved in handover by moving across different APs within their geographical coverage. Handovers can either happen in intra- or inter-zone mode. Inter-zone handovers occur when a node moves across different zones served by different APs, whereas intra-zone

Digital Object Identifier:
10.1109/MCOM.2018.1700863

Vishal Sharma and Ilsun You (Corresponding author) are with the Soonchunhyang University; Francesco Palmieri is with the University of Salerno; Dushantha Nalin K. Jayakody is with the National Research Tomsk Polytechnic University; Jun Li is with the Nanjing University of Science and Technology.

0163-6804/18/$25.00 © 2018 IEEE

handovers take place within the same zone but with different APs. Intra-zone handovers are easier to manage, and their security can be handled by existing Proxy-Mobile IPv6 (PMIPv6)-based solutions. However, inter-zone handover requires seamless connectivity without compromising the security of the network. Over the years, several solutions have been proposed within the PMIPv6 framework for such inter-zone handovers. However, these solutions defy the core concept of flattened architecture and may introduce excessive signaling overheads in fog networks enabling IoT services. Some scenarios may suggest the introduction of local authentication servers, but such servers will also heavily depend on a centralized authentication one for periodic updates and validations.

Another major issue with handover is energy consumption. In a complex architecture, security involves many entities, which increases the amount of signaling messages needed by adversely affecting the devices' energy budget. Overconsumption of energy also happens while maintaining uplinks/downlinks between entities during handover. Recent studies (e.g., [1–3]) have discussed the impact of signaling overhead on energy consumption. Thus, enhancing handover security without many dependencies on hierarchical authentication mechanisms and with less signaling burden can also improve energy efficiency. These hierarchical authentication procedures can be attained by distributed mobility management (DMM), as explained in the next section.

Distributed Mobility Management in Fog Networks

Hierarchical dependencies within networks introduce several problems related to latency and communication overhead. In the presence of high mobility, things are worsened due to limited control over MNs. DMM has been introduced to support low latency and flat architectures despite the presence of hierarchical dependencies. It is aimed at distributing mobility control across the network over multiple different entities, avoiding any centralized mobility management mechanism (https://tools.ietf.org/html/draft-chan-dmm-distributed-mobility-anchoring-01). In order to reduce network load by better balancing it, DMM anchors the traffic near mobile and service nodes (https://tools.ietf.org/html/rfc7333).

DMM can be structured according to a semi-distributed or fully distributed model (https://tools.ietf.org/html/draft-wt-dmm-deployment-models-00). In the former, there is a centralized controller, whereas the latter implements distribution strategy even for the controller. Intelligent service functions should be developed that can be converted into semi-distributed or fully distributed solutions without much overhead. DMM can also be implemented as a host-based or network-based solution. Host-based solutions allow MNs to actively participate in the mobility management and handoffs, whereas network-based solutions do not allow active involvement of MNs in mobility management.

The implementation of state-of-the-art DMM solutions in the fog network scenario is sketched in Fig. 2 and discussed in the following.

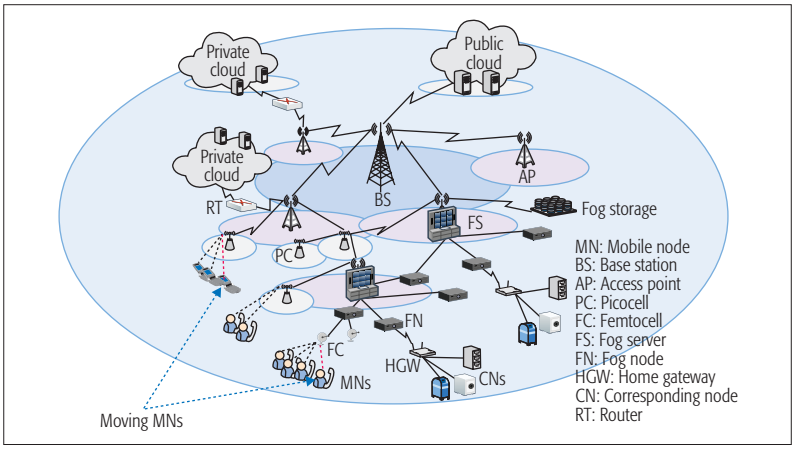

Figure 1. An example of fog network architecture.

PMIPv6-Based DMM: PMIPv6 relies on Proxy Binding Update (PBU) and Proxy Binding Acknowledgment (PBA) messages between the local mobility anchor (LMA) and mobility access gateways (MAGs). The LMA is the central authority that manages the movement of MNs across the MAGs. The CNs in PMIPv6 are governed by their HGWs, and the link for communication is formed as MN-MAG-LMA-HGW-CN. PMIPv6-based DMM replaces centralized LMA with a control mobility database (CMD) that regulates the mobility rules for mobility anchors and access routers (MAARs), which are the replacement for MAGs. According to the reports on PMIPv6-based DMM, CMD defines the session updates depending on its uses as a PBU/PBA relay, MAAR locator, or MAAR proxy [4, 5]. An example scenario is shown in Fig. 2a with its corresponding protocols reported in Figs. 3a-1 and 3a-2. We consider two PMIPv6-based DMM variants presented in [4, 6] that differ in message exchange but both rely on the creation of a tunnel between the MAARs. PMIPv6 is able to withstand security issues within the handover; however, it also suffers from some vulnerabilities and can expose the network to intruders during tunnel setup. For example, backward broadcasting can result in capturing traffic on a compromised MAAR. Furthermore, PMIPv6-DMM security depends on the assumption that CMD is not compromised. This assumption is too strong as CMD is exploitable by attackers in several scenarios. Apart from these issues, the hierarchical procedures make de-registration insecure and can lead to the disclosure of MN information.

Software Defined Networking (SDN)-Based DMM: SDN-based DMM uses the SDN controller as a centralized FS, which makes the resulting architecture semi-distributed. SDN-based DMM leverages the flow table architecture of different variants of SDN technology [7] [8]. There are different SDN-based DMM schemes, but we consider route-optimization-based SDN-DMM as shown in Fig. 2b [9]. This scheme copes with route optimization issues in SDN-based DMM by improving the path of the previous flow, which is managed by the source MAAR. Currently, there is no support for security. However, as shown in Fig. 2, this scheme can be improved by adding a security server along with the controller. From Fig. 2b, it can be seen that the controller performs the tasks of CMD and uses binding updates to manage the

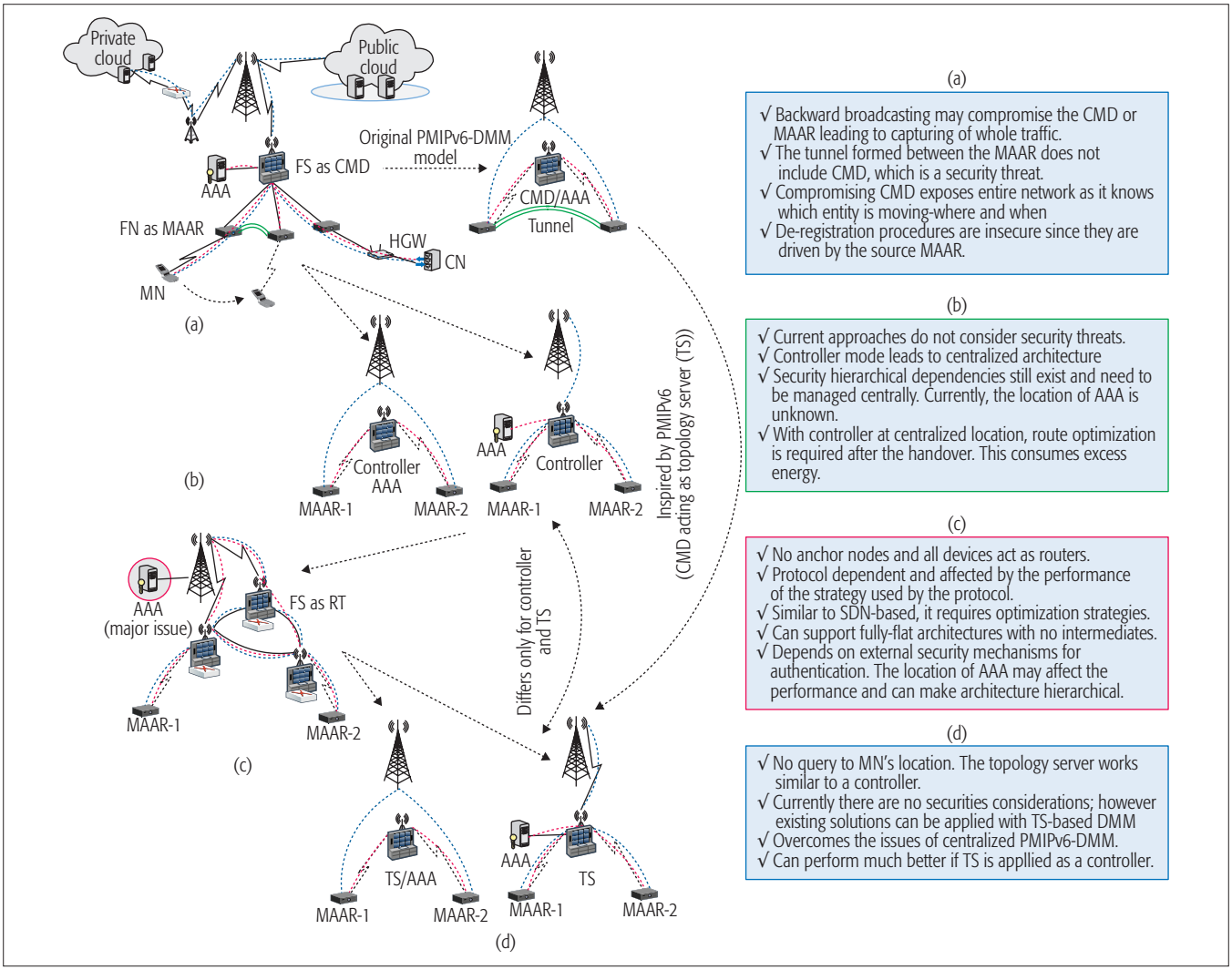

Figure 2. An illustration of exemplary scenarios of fog networks using PMIPv6, SDN, routing, and topology-based DMMs: a) PMIPv6-based DMM; b) SDN-based DMM; c) routing-based DMM; d) topology-based DMM.

handovers. Before handovers, the controller uses binding cache entry (BCE) and modifies the flow messages. Unlike PMIPv6-based DMM, SDN-based DMM relies on router update messages for shifting controls between the MAARs. All the information about the MN is passed to the new MAAR via the controller, introducing excessive routing burden on the network. The variant presented in Fig. 2b achieves route optimization by modifying the flow tables. These modifications are performed by the controller depending on the network flow in order to allow direct messaging between the HGW of the CN and the target MAAR.

Routing-Based DMM: This allows fully distributed mobile network architectures and also paves the way to flat security solutions. However, routing-based DMM has to rely on a centralized authentication server for governing local authentication policies. This may perceivably affect the performance depending on routing and optimization issues. Figures 2c and 3c present the Border Gateway Protocol (BGP)-based DMM scheme [10] as considered in [5] using the FS as a router (RT). Unlike other DMM solutions, this is fully distributed, but its flat architecture gets affected by the introduction of security mechanisms. Routing-based DMM is protocol-dependent and relies on Domain Name Server (DNS) queries, and uses router advertisements and authentication messages for connections. The major disadvantage is that it depends on external security procedures, which introduce excessive performance overheads.

Topology-Based DMM: This is a variant of SDN-based DMM, which relies on policies driven by PMIPv6-based DMM [11]. Instead of working as a controller, the FS operates as a topology server (TS) and contributes to mobility anchoring as shown in Fig. 2d. This DMM solution can be useful in semi-distributed scenarios and can prevent backward looping as no query is required on an MN's location during handover. It uses a combination of Dynamic Host Configuration Protocol (DHCP) requests and PBU/PBA messages to establish a tunnel between the previous and target MAARs. However, considering the consequences of PMIPv6-based DMM, topology-based DMM also suffers from a similar issue of backward broadcasting, and its security follows a hierarchical scheme. In the current Internet Engineering Task Force (IETF) draft on topology-based DMM, there are no security considerations; however, we emphasize its importance considering that the previous MAAR does not need to inform

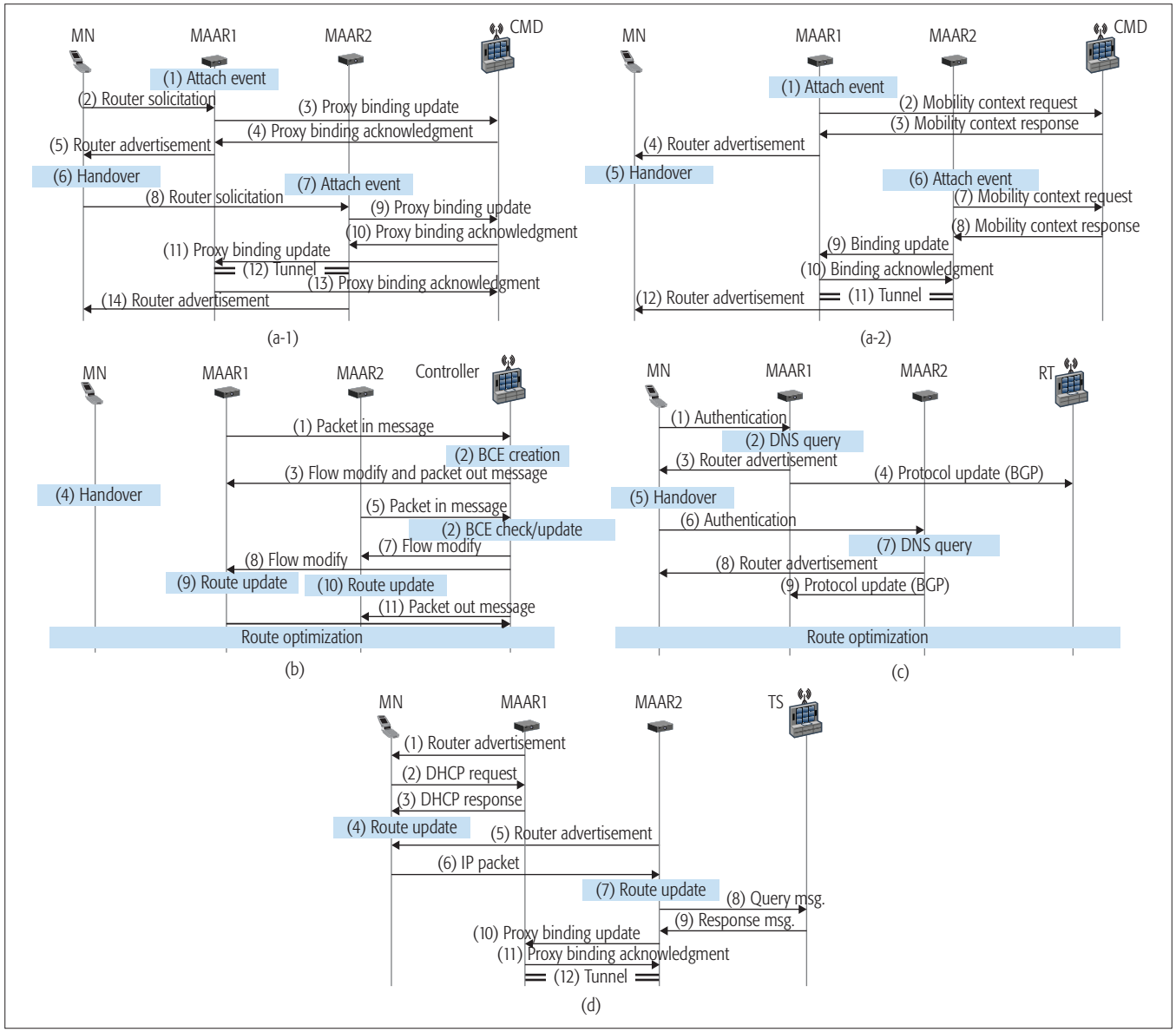

Figure 3. An illustration of the message exchange mechanism of PMIPv6, SDN, routing, and topology-based DMMs in fog-IoT networks.

the target MAAR about the location of MNs and vice versa.

DMM resolves many hierarchical management issues by flattening the network; however, security layout remains hierarchical. A comparison between the available solutions is shown in Fig. 4a. SDN- and topology-based DMMs can provide better management and control with lower scalability overheads, whereas PMIPv6-based DMM can overcome issues related to signaling overheads and security. Routing-based DMM can provide a non-ownership type of network, which can be configured without changing any signaling messages between the entities. However, all of the above DMM implementations suffer from resource inequality and identity management issues for practical applications. Out of these solutions, PMIPv6-based DMM has proven to be competitive in terms of security. Thus, threat implications are presented around this type of DMM, as shown in Fig. 4b.

In PMIPv6-based DMM, the communication between MN and MAAR is subject to session hijacking and denial of service attacks, which can be hindered by mutual authentication and message protection. The links between CMD and MAAR can be disrupted as a consequence of attacks launched at the MAAR or CMD level. The two existing variants for PMIPv6-based DMM [4, 6] can be leveraged to face these issues, but only to a limited extent. Of both solutions, the one proposed in [6] reduces the burden of CMD and is well aligned with the objectives of DMM. Signaling overhead in [4] is high as the CMD sends PBUs directly to the previous MAAR rather than to the local area network. Further, the variant in [6] can be trusted because the serving MAAR directly contacts other involved ones after receiving the PBU from CMD. Thus, it is difficult for the compromised CMD to deceive other MAARs. On the other hand, variants in [4, 5] are vulnerable to attacks by the compromised CMD because it can directly contact the involved MAARs. Moreover, the tunnel created between both MAARs is subject to threats despite the fact that the previous MAAR sends PBA after tunnel formation

The security problems in PMIPv6-based DMM can be solved with a more complicated authentication with the centralized server. However, this will break the objectives of DMM, by making the network hierarchical and introducing more signaling overhead, and consequently consuming more energy.

Metrics	PMIPv6-based DMM	SDN-based DMM	Routing-based DMM	Topology-based DMM	Blockchain-based DMM
Security considerations	Weak (via IPv6 tunneling)	No (requires external mechanisms)	No (requires external mechanisms)	No (requires external mechanisms)	Yes (distributed)
Architecture/mode/ security layout possibilities	Hierarchical to flat/semi-distributive /hierarchical	Hierarchical to flat/semi-distributive /hierarchical	Hierarchical to flat/fully-distributive /hierarchical	Hierarchical to flat/semi-distributive /hierarchical	Hierarchical to flat/fully-distributive /distrubutive
Fault-tolerance	Partial (tunnel dependent)	High (controller dependent)	Partial (protocol dependent)	High (TS dependent)	High (individual dependent)
Control and management	Low	High	Low	Moderate	High
Single point of failure	Yes	Yes	No	Yes	No
De-registration security	No	No	No	No	Yes
MN location query	Required	Not required	Required	Not required	Not required
Scalability to network densification	Low	Moderate	Low	Moderate	High
Scalability overheads	High	Low	High	Low	Moderate (redundancy)
Identity management and protection	No	No	No	No	Yes
Resource inequality	High	Moderate	High	High	Low
Ownership	CMD	Controller	None	TS	Genesis (parent)

(a)

Type	Communication entity	Message	Security threat	Effect	Countermeasures
PMIPv6-based DMM	MN - MAAR	Router solicitation	Session hijacking	The attacker hijacks the session by impersonating the victim node.	Message protection and mutual authentication
		Router advertisement	Denial of service	The attacker node can set the wrong network information.	Message protection
	CMD - MAAR	PBU / PBA MCReq/MCRes	Attack by malicious MAAR	√These messages are protected, but if one MAAR is compromised, various attacks are possible through it. √Compromised MAAR can redirect traffic, which is coming from other MAARs. √The attacks directly launched by the compromised MAAR cannot be prevented. In other words, when the primary defense is broken, it is important to protect the other involved MAARs from being affected by the compromised one. If the MAAR deceives other entities with the false arrival of MN: >Bernardos et al. (2012)/Giust et al.(2015): The MAAR falsifies CMD to send the wrong PBU to the previous MAARs. >Lee et al. (2016): By directly sending the wrong PBUs, the new MAAR attempts to trick the previous MAARs in addition to CMD.	Handover will of the MN should be verified separately.
			Attack by malicious CMD	If CMD is compromised, it attempts to launch the redirect attach by sending the false PBUs to the MAARs	
	MAAR - MAAR	BU / BA	Attack by malicious MAAR	√These messages are protected, but if one MAAR collapses, various attacks are possible through it. If the MAAR deceives other entities with the false arrival of MN: >Bernardos et al. (2012)/Giust et al. (2015): The malicious MAAR tricks CMD for sending the wrong PBU to previous MAARs. >Lee et al. (2016): After MAAR identifies that the CMD receives information about the previous MAARs, it directly sends PBUs to them. The MAAR tries to trick the previous MAARs in addition to CMD.	Handover will of the MN should be verified separately.

(b)

Figure 4. Comparison of PMIPv6, SDN, routing, and topology-based DMMs and threat implications of PMIPv6-based DMM in fog networks.

as shown in [4, 5]. This is because there are no de-registration policies that mean a compromised previous MAAR can itself respond to the PBU without letting a target MAAR know about its situation and thus expose the network. The security problems in PMIPv6-based DMM can be solved with a more complicated authentication with the centralized server. However, this will break the objectives of DMM by making the network hierarchical and introducing more signaling overhead, and consequently consuming more energy.

A Fully Distributed Solution for Managing Security: The Blockchain

Most of the aforementioned problems are associated with the need to centralize the control of security data, while managing DMM in a distributed scenario. Centralized data recording is tedious, is often inefficient, introduces a single point of failure, and requires complex procedures to prevent overriding and overlapping while committing an action. Operating databases in a distributed manner resolves this problem, but introduces other issues related to authenticating and trusting the data managing entities. One of the most interesting solutions for coping with such requirements is blockchain, as defined by Satoshi Nakamoto in 2008, which is based on a trustless consensus scheme.

In the original blockchain architecture, the effectiveness of such a scheme is ensured by a proof of work (PoW) mechanism that, for the generation of a new block, requires solving a very complex and computationally expensive mathematical puzzle. In contrast, the verification process is extremely simple and typically associated with a single hashing operation. Blockchain's architecture allows managing records in peer-to-peer format by keeping information in blocks and ensuring the complete control and verifiability of each transaction. These can be seen as automatically growing programmable ledgers [12].

The blockchain is secured by its design and rules allowing Byzantine fault tolerance. The blockchain operates by distributing copies of existing records as electronic ledgers to all the participants that either add or delete a record to maintain the information. In-place amendments of records are not allowed in the blockchain. The parent node of the blockchain is termed the "genesis" followed by "main chain" and "orphans." Generally, blockchain uses a public key as the address of each block and a private key as the pass key to access critical information. However, the initial concept focuses only on the public key from the point of making blockchain openly accessible, permissible, and non-forkable [12, 13].

The key components of an efficient blockchain include:

- *Electronic ledger*, which is shared among all peers
- *Smart contract*, which defines the terms, rules, and conditions for accessing ledgers via transactions
- *Privacy and validations*, which form the security concepts and procedures for verifying transactions
- *Protocol and application interface*, which sets operation rules and accessibility solutions for ledgers

Since blockchain conceptualizes the removal of a centralized entity for security, it can serve the purpose of maintaining a flat and distributed layout in highly dense fog networks. For further details, readers can see [12–14].

Blockchain-Based DMM for Secure and Energy-Efficient Handovers

We propose a novel blockchain-based DMM scheme capable of overcoming the distributed security concerns of the existing solutions. Three different blockchains are used, namely, PoW-wise, region-wise, and user-wise ones. The PoW-wise blockchain is the private blockchain implemented at the FS level in the fog network. This is responsible for coordinating inter-MAARs ledgers and supporting conflict issues between the MAARs.[1] The conflict issues are stored as a value representing the number of times a handover between the entities failed. The region-wise blockchain controls the MAARs, and the user-wise blockchain comprises MNs. There are three different block formats for the three different blockchains as shown in Figs. 5a and 5b. All the entities are identified by their physical address (ID) and public key (PK). The MN-block format uses a hash for its own and previous members of a user-wise blockchain as in traditional blockchain schemes. It is capable of generating signatures on the basis of hashes and stores the conflict value for any of the irregular MNs with the timestamp (t). The timestamp helps to identify the blockchain properties and state at the time when a particular conflict occurs. MNs use an access key (X) from the target MAAR in the case of inter-zone handover, which obtains X from the PoW (FS) access key pool. Alternatively, this access key can only be generated by MAARs and does not need to be obtained from any PoW-FS node. The MN stores the block strength of the user-wise blockchain and keeps track of it via an update counter. The MAAR block format is a bit more complex with entries governing MAARs, the hash for connected MAARs, block strength of MNs, conflict values for connected MAARs, handover traces, access keys for PoW ledgers, and displayed key for other MAARs necessary for identifying the source PoW node. Similarly, PoW node block format contains information about other PoWs with a list of MAARs and their strengths.

The MN traffic is anchored by its serving MAAR toward the HGW of a CN, and in this case, the traffic arrives at the serving MAAR when the MN is in transit toward its target MAAR; the serving MAAR communicates directly with the target MAAR through an FC or a PC by identifying the location of the MN through region-wise ledgers. However, if the MN is untraceable from region-wise ledgers, FS is used for anchoring the traffic.

In the initial phase,[2] the MN and the serving MAAR exchange a blockchain handover key (BCHK) for securing their communication. Further, the MN shares a one-time password (OTP) with the serving MAAR, which is generated as an *N*-hash chain of OTPs with a key (K) similar to the mechanism of S/Key (https://tools.ietf.org/html/rfc1760). This S/Key-OTP fortifies a lightweight authentication, which can avert a network by an interceptor who performs perpetual sniffing.

In the original blockchain architecture, the effectiveness of such scheme is ensured by a Proof of Work (PoW) mechanisms, that, for the generation of a new block requires solving a very complex, and computationally expensive, mathematical puzzle. In contrast, the verification process is extremely simple and typically associated with a single hashing operation.

[1] MAAR 1 and MAAR 3 are the serving MAARs, and MAAR 2 and MAAR 4 are the target MAARs.

[2] It is assumed that mutual authentication and key exchange procedures are performed for safely sharing the BCHK and the first OTP. However, work is still required in this direction for ensuring the security during the initial phase of handover.

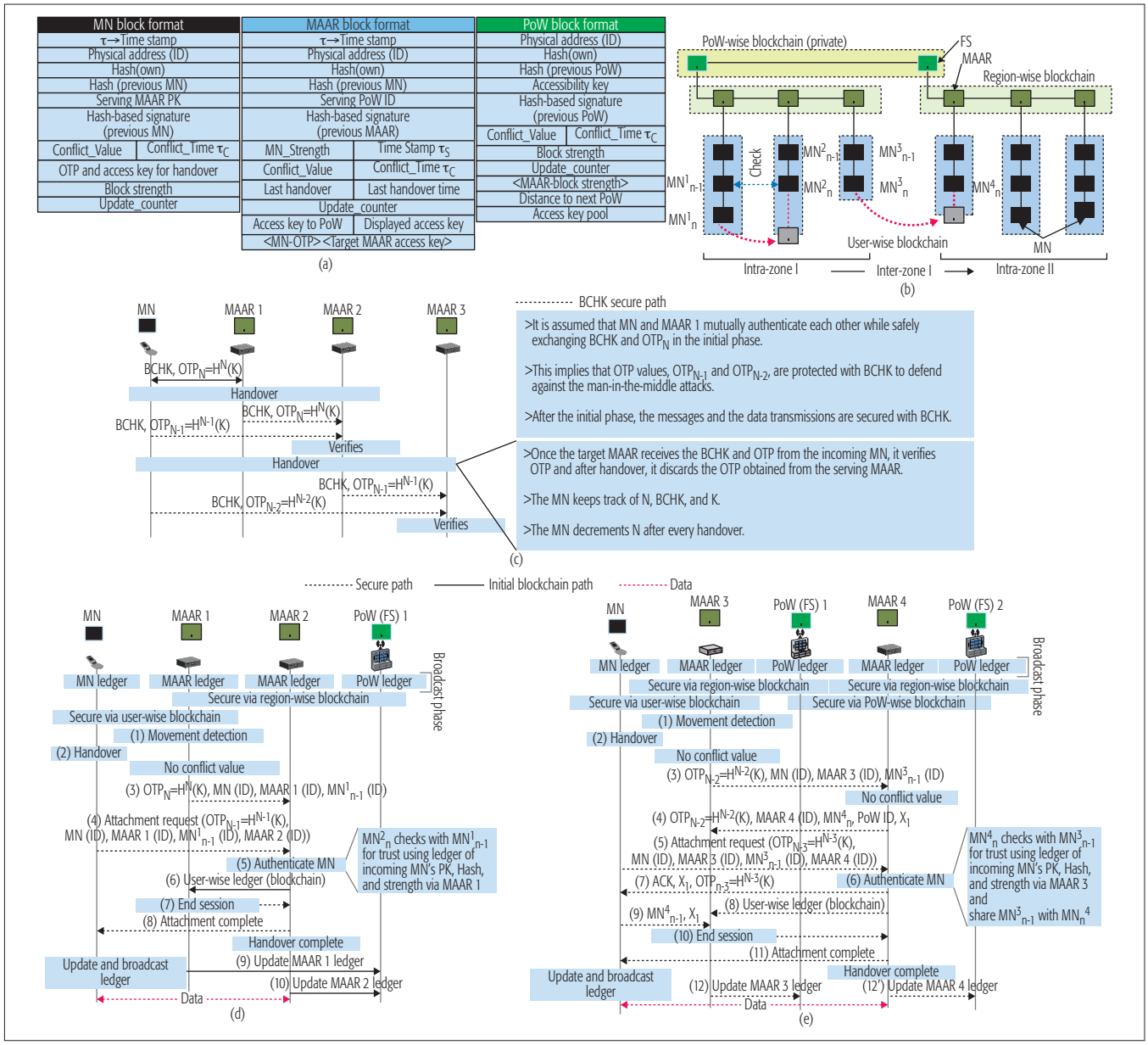

Figure 5. a) Block format for MN, MAAR, PoW (FS); b) DMM handover scenario with user-wise blockchain, region-wise blockchain, and PoW-wise blockchain; c) an initial phase and handover overview; d) intra-zone handover between MAARs of same PoW (FS) using blockchain-based DMM; e) inter-handover between MAARs of different PoWs (FS) using blockchain-based DMM.

The OTP is passed by the serving MAAR to the target MAAR, which also obtains the next OTP from the MN and verifies it followed by other procedures for managing handover, as shown in Fig. 5c. Although a compromised MAAR exposes OTP as well as BCHK, it cannot affect the network as the authentication procedures are secured via three blockchain ledgers.[3]

For intra-zone handover, as shown in Fig. 5d, the process starts with the sharing of ledgers of the three blockchains, provided according to a broadcasting strategy.[4]

(1)–(2): Once the serving MAAR identifies the movement, the procedures for handover are initiated.

(3): The serving MAAR checks for previous conflicts for this MN, and if no conflict is identified, it sends a handover message including $OTP_N = H^N(K)$, IDs for fetching public keys (PKs) of the MN, MAAR 1, and MN^1_{n-1} to the target MAAR.

(4): The MN proceeds with the attachment request with a similar context along with ID of target MAAR with $OTP_{N-1} = H^{N-1}(K)$.

(5): On receiving the request, the target MAAR verifies the $OTP_{N-1} = H^{N-1}(K)$ with $OTP_N = H^N(K)$. Next, it accounts for the user-wise ledger of MN^2_n, which obtains information from the MN^1_{n-1} regarding the hash and strength, and uses it to validate the incoming MN. Unlike existing DMM, there is no CMD involved, and validation procedures are carried out using the distributed nature of blockchains.

(6)–(8): Once authorized, the target MAAR sends a user-wise ledger to the serving MAAR, which analyzes it for any suspicious activity. It sends an end-session message if everything is fine, and the target MAAR sends an attachment complete message to MN, thus completing the handover.

[3] In addition, BCHK should not be directly sent from the MN to MAAR 2 or MAAR 3. Instead, it is safely forwarded to MAAR 2 and then to MAAR 3 via secure channels between MAAR 1 and MAAR 2, and between MAAR 2 and MAAR 3, respectively.

[4] Currently, no specific broadcasting scheme is considered for blockchain-based DMM, and it is an open issue.

(9)–(10): Finally, the MN updates and broadcasts the user-wise ledger. The involved MAARs share their region-wise ledgers with the updated strength and information to the PoW node (FS).

For inter-zone handover operations, all three blockchains are involved, whereas only two are used in intra-zone mode, as shown in Fig. 5e. The initial steps are similar to (1)–(2) of intra-zone mode, followed by the following steps:

(3): The movement detection is followed by a handover message from the serving MAAR to the target MAAR including $OTP_{N-2} = H^{N-2}(K)$, IDs for fetching the PK of itself, MN, and MN^3_{n-1}, which is the previous MN of the serving MAAR.

(4): The target MAAR uses PoW-wise blockchain and checks for the existence of a conflict with the serving MAAR. If no conflict exists, it sends an access key (X_1) along with $OTP_{N-2} = H^{N-2}(K)$, *ID* of itself, MN^n_4, and PoW (FS) ID.

(5): The MN sends an attachment request to the target MAAR using $OTP_{N-3} = H^{N-3}(K)$ along with IDs for fetching the PK of MN, MAAR 3, MAAR 4, and MN^3_{n-1}.

(6)–(7): Following the authentication procedure as in intra-zone step (5), the MN receives an acknowledgment including X1 and $OTP_{N-3} = H^{N-3}(K)$. The MN retrieves MN^4_n, which now becomes MN^4_{n-1} from the target user-wise ledger.

(8)–(9): The target MAAR sends an updated user-wise ledger to the serving MAAR, which also receives the context information from the MN with X_1 and MN^4_n The serving MAAR checks for any suspicious activity before ending the session.

(10)–(11): On receiving the end session message from the serving MAAR, the target MAAR informs the MN about the completion of attachment procedures, and handover is completed.

(12)–(12′): Finally, the involved MAARs update their region-wise ledgers to their respective PoW-FS nodes.

The blockchain-based DMM overcomes the threat implications of PMIPv6-based DMM and also resolves the issues related to de-registration and backward broadcasting leading to tunnel attacks. The next section presents the performance evaluation of the proposed solution.

Performance Evaluation

The signaling overhead associated with the registration of an MN into the blockchain of a target MAAR (inter-zone mode) is analyzed by comparing it to the host-based as well as network-based DMMs [6]. We used an evaluation methodology similar to the one presented in [6], where the signaling overhead (SO) associated with registration is given as

$$\mathcal{S}_{\mathcal{O}} = \frac{\mathcal{B} \times \mathcal{M}}{\mathcal{T}} + \mathcal{A}. \quad (1)$$

Here, $\mathcal{B}$ is the number of hops between MN and MAAR or PoW-FS (10 for network-based and 1 for host-based), $\mathcal{M}$ is the registration packet length (66 bytes), and $\mathcal{A}$ is the excessive overhead given as

$$\frac{(\mathcal{B}-1)\times\mathcal{M}}{\mathcal{T}} + \frac{(k-1)\times(\mathcal{B}')\times\mathcal{M}}{\mathcal{T}}$$

for network-based and as

$$\frac{(k-1)\times(\mathcal{B}')\times\mathcal{M}}{\mathcal{T}}$$

for host-based, where $\mathcal{B}'$ is the hop distance between APs (with a value of 3), and k is the number of addresses configured for an MN [6]. Although the proposed approach can be more aggressive on memory consumption because of redundant operations of the blockchain, it does not affect the signaling messages. On the contrary, it reduces the number of addresses to be configured at the MN and also considerably decreases the distance between the hops as the user-wise blockchain is updated in the majority of the scenarios, which follows one-hop distance. Only two addresses are associated with the MN at any given time because of the concept of the blockchain. Thus, the percentage improvement for the blockchain-based DMM in comparison to the network-based DMM and host-based DMM (at $k = 5$) by varying $\mathcal{T}$, which is the stay time of an MN, is 86.6 and 69.2 percent, respectively, as shown in Fig. 6a. The results will change once the redundant message size is considered along with the amendment overheads of the blockchain. However, its impact can be neglected considering the distributed security of the proposed blockchain-based DMM.

The proposed blockchain-based DMM and host-based DMM strategies both imply per-node transmissions $\mathcal{C}_{tx}$ and receptions $\mathcal{C}_{rx}$ in the ratio of 2:2, thus, they consume similar amounts of energy, whereas network-based DMM has these in the ratio 1:2. This energy can be calculated as

$$\varepsilon = (\mathcal{C}_{tx} \times T_X \times \mathcal{I}_t) + (\mathcal{C}_{rx} \times R_X \times \mathcal{I}_t) + (\mathcal{P}_{rp} \times (\mathcal{T} - \mathcal{I}_t)), \quad (2)$$

where T_X is the transmission power (1726 mW), R_X is the reception power (1340 mW), $\mathcal{I}_t$ is connection inactivity time (0.1–0.3 s), and $\mathcal{P}_{rp}$ is the reception power in the zone of the MAAR (1325 mW) [15]. For the entire network, this energy consumption becomes

$$\varepsilon_{\mathcal{N}} = \left(\left(\left(\frac{\mathcal{S}_{\mathcal{O}}}{\mathcal{B}} \times \alpha_1\right) + \alpha_2\right) \times \mathcal{I}_t \times \left(\mathcal{C}_{tx} + \mathcal{C}_{rx}\right)\right) + \left(\mathcal{P}_{rp} \times \left(\mathcal{T} - \mathcal{I}_t\right)\right), \quad (3)$$

where α_1 and α_1 are the power constants with values 438 mW per unit byte per second and 1288 mW, respectively [15] As expected, an MN in network-based DMM consumes lower energy than the host-based DMM and blockchain-based DMM; however, the difference is marginal (i.e., only 1 percent). For the overall network, the proposed blockchain-based DMM consumes 4.7 percent more energy than the network-based DMM, but 17.5 percent lower than the host-based DMM, as shown in Fig. 6b The results are also recorded by following the one-hop concept of blockchain-based DMM that relies only on broadcasting the ledgers without relaying, thus reducing the overall hop distance. The results in Fig. 6c suggest that one-hop implementation reduces the energy consumption up to 3.5 per-

Although the proposed approach can be more aggressive on memory consumption because of redundant operations of the blockchain, yet it does not affect the signaling messages. On the contrary, it reduces the number of addresses to be configured at the MN and also considerably decreases the distance between the hops as the user-wise blockchain is updated in the majority of the scenarios, which follows one-hop distance.

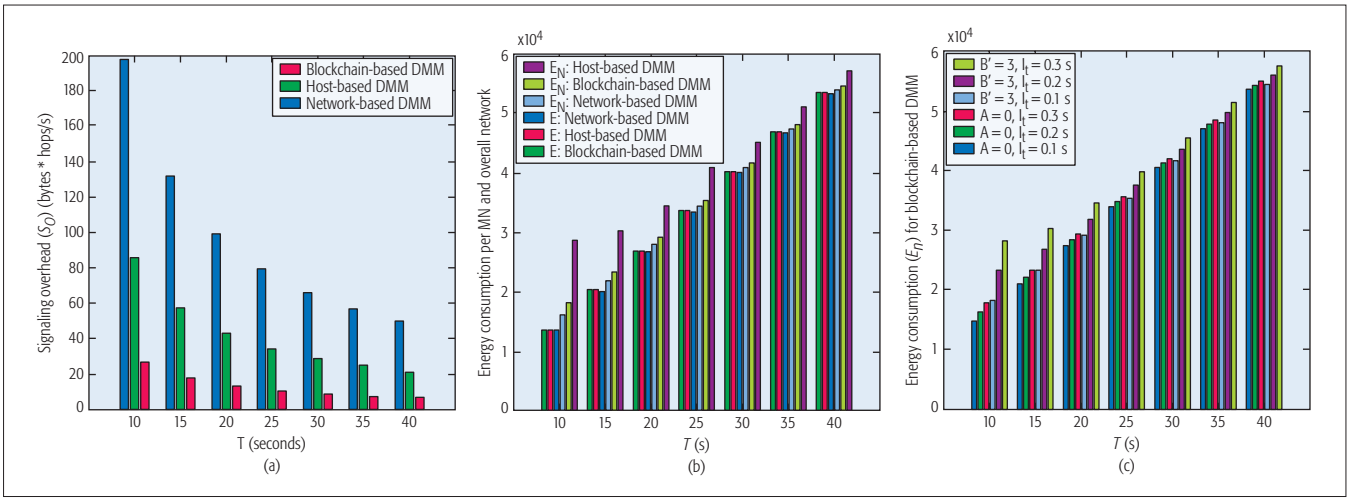

Figure 6. Simulation results: a) $\mathcal{S}_{\mathcal{O}}$ vs. duration of MN in the network; b) energy consumption vs. duration of an MN in the network with varying $\mathcal{S}_{\mathcal{O}}$; c) energy consumption for one-hop and $\mathcal{B}$-hop blockchain-based DMM with variable $\mathcal{I}_t$.

cent for $\mathcal{I}_t$ at 0.1, 0.2, and 0.3 s. However, it will increase because of redundant operations of the blockchain. At present, these evaluations do not include energy consumed in the broadcasting of ledgers. This will surely increase the energy consumption and will depend on the consensus of broadcasting scheme, but considering the advantages and security of blockchain-based DMM, it can be relaxed at the moment. Currently, the blockchain formation time and memory requirements are not considered. As there is no standard format for a packet header to be used in the case of blockchain-based DMM, it can be relaxed at the moment.[5]

Conclusions and Future Remarks

This work proposes a new DMM solution for flattened fog network architectures leveraging the blockchain technology. The proposed blockchain-based DMM is capable of coping with hierarchical security issues without affecting the network layout. It satisfies the requirements of fully distributed security by using three blockchains and also resolves the de-registration issues affecting the existing DMM solutions. It also significantly reduces signaling burden compared to the existing DMM solutions, resulting in reduced energy consumption.

The proposed blockchain-based DMM is capable of resolving issues related to defense against malicious MAARs, malicious CMDs, and malicious MNs. Further, the distributed blockchain strategy helps to counterfeit attacks, such as denial of service/distributed denial of service, impersonation, session hijacking, and backward broadcasting. Also, the proposed solution supports de-registration policies, all of which are major security issues of the existing PMIPv6-based DMM. In spite of such extensive resolutions, there are various issues that need to be taken care of while further enhancing the blockchain-based approach. These include solutions for redundant memory requirements. The multiple ledgers may use excessive memory, which may affect the performance. Thus, controlled operations must be performed for memory consumption. Initial authentication for blockchain-based DMM is based on the assumption of a secure channel. This assumption should be relaxed with a novel ideology for a secure initial communication. Solutions for session key exchange along with privacy and anonymity have yet to be explored for blockchain-based DMM. Apart from these, there is no standard packet format for exact implementation of blockchain-based DMM, which should be standardized along with the broadcast consensus scheme and the blockchain formation time.

In future research directions, the proposed blockchain-based DMM will be evaluated with different types of broadcasting mechanisms. In the presence of memory constraints, it is mandatory to resolve multi-blockchain dependencies. This can be performed by moving toward network architectures with no managing authority. However, these architectures also suffer from other limitations such as service charging policies, record management, and tracking, and hence require further exploration.

Despite being in an early stage, the proposed blockchain-based DMM has been demonstrated to be extremely promising and can be seen as one of the key solutions for secure distributed handover in fog networks.

[5] Currently, the blockchain formation time and memory requirements are not considered as there is no standard format for a packet header to be used in the case of blockchain-based DMM.

Acknowledgment

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2016R1D1A1B03935619) as well as by the Soonchunhyang University Research Fund. Ilsun You is the corresponding author.

References

[1] B. Klaiqi, X. Chu, and J. Zhang, "Energy-Efficient and Low Signaling Overhead Cooperative Relaying with Proactive Relay Subset Selection," *IEEE Trans. Commun.*, vol. 64, no. 3, 2016, pp. 1001–15.
[2] Y. Song, W. Choi, and S. Baek, "Network Switching Strategy for Energy Conservation in Heterogeneous Networks," *PloS one*, vol. 12, no. 2, 2017, p. e0172318.
[3] K. F. Doppler *et al.*, "Method and Apparatus for Power Saving Operations in Wireless Network Elements," June 27, 2017, U.S. Patent 9,693,299.
[4] C. Bernardos *et al.*, "A Pmipv6-Based Solution for Distributed Mobility Management," IETF draft-bernardos-dmm-pmip- 01; https://tools.ietf.org/html/draft-bernardos-dmm-pmip- 01, accessed Jan. 8, 2018.

[5] F. Giust, L. Cominardi, and C. J. Bernardos, "Distributed Mobility Management for Future 5G Networks: Overview and Analysis of Existing Approaches," *IEEE Commun. Mag.*, vol. 53, no. 1, Jan. 2015, pp. 142–49.
[6] J. h. Lee *et al.*, "Distributed IP Mobility Management from the Perspective of the IETF: Motivations, Requirements, Approaches, Comparison, and Challenges," *IEEE Wireless Commun.*, vol. 20, no. 5, Oct. 2013, pp. 159–68.
[7] T.-T. Nguyen, C. Bonnet, and J. Harri, "SDN-Based Distributed Mobility Management for 5G Networks," *Proc. IEEE WCNC*, 2016, pp. 1–7.
[8] Y.-H. Kim *et al.*, "A Sdnbased Distributed Mobility Management in LTE/EPC Network," *J. Supercomputing*, vol. 73, no. 7, 2017, pp. 2919–33.
[9] H. Yang and Y. Kim, "SDN-Based Distributed Mobility Management," *Proc. Int'l. Cong. Info. Networking*, Jan. 2016, pp. 337–42.
[10] P. McCann, "Authentication and Mobility Management in a Flat Architecture," IETF draft-mccann-dmmflatarch- 00, 2012' https://tools.ietf.org/html/draft-mccanndmm- flatarch-00, accessed Jan. 8, 2018.
[11] J. Lee and Y. Kim, "Topology-Based Distributed Mobility Anchoring in Pmipv6," IETF draftjaehwoon- dmm-topology–mobility-anchoring-00, 2016; https://tools.ietf.org/html/draft-jaehwoon-dmm-topology– mobility-anchoring-00, accessed Jan. 8, 2018.
[12] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," *IEEE Access*, vol. 4, 2016, pp. 2292–2303.
[13] F. Tschorsch and B. Scheuermann, "Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies," *IEEE Commun. Surveys & Tutorials*, vol. 18, no. 3, 2016, pp. 2084–2123.
[14] A. Stanciu, "Blockchain Based Distributed Control System for Edge Computing," *Proc. 21st IEEE Int'l. Conf. Control Systems Computer Science*, 2017, pp. 667–71.
[15] G. Foddis *et al.*, "LTE Traffic Analysis for Signalling Load and Energy Consumption Trade-Off in Mobile Networks," *Proc. IEEE ICC*, 2015, pp. 6005–10.

Biographies

Vishal Sharma [S'13, M'17] received his Ph.D. and B.Tech. degrees in computer science and engineering from Thapar University (2016) and Punjab Technical University (2012), respectively. He worked at Thapar University as a lecturer from April 2016 to October 2016. He was a postdoctoral researcher at Soongsil University and Soonchunhyang University, South Korea, from November 2016 to September 2017. Currently, he is a research assistant professor in the Department of Information Security Engineering, Soonchunhyang University.

Ilsun You [SM'13] received his M.S. and Ph.D. degrees in computer science from Dankook University, Seoul, Korea, in 1997 and 2002, respectively. He received his second Ph.D. degree from Kyushu University, Japan, in 2012. From 1997 to 2004, he was at THIN Multimedia, Internet Security, and Hanjo Engineering as a research engineer. Now, he is an associate professor in the Information Security Engineering Department, Soonchunhyang University. He is a Fellow of the IET.

Francesco Palmieri [M'17] is an associate professor in the Computer Science Department of Salerno University. He received his M.S. (Laurea) degree and Ph.D. in computer science from Salerno University. He has been closely involved with the development of the Internet in Italy as a senior member of the Technical-Scientific Advisory Committee and of the CSIRT of the Italian NREN GARR.

Dushantha Nalin K. Jayakody [M'14] received his B. Eng. degree in Pakistan. He received his M.Sc. degree in electronics and communications engineering from Eastern Mediterranean University, Cyprus. He received his Ph. D. degree in electronics and communications engineering from University College Dublin and held a postdoctoral position (2014–2016) at the University of Tartu and the University of Bergen. Now, he is an associate professor at the Institute of Cybernetics, National Research Tomsk Polytechnic University, Russia.

Jun Li [M'09, SM'16] received his Ph. D degree in electronic engineering from Shanghai Jiao Tong University, P. R. China, in 2009. Since June 2015 he has been a professor at the School of Electronic and Optical Engineering, Nanjing University of Science and Technology, China. Before this, he was a research scientist (2009) at Alcatel Lucent Shanghai Bell, a postdoctoral fellow (2009–2012) at the University of New South Wales, Australia, and a research fellow (2012–2015) at the University of Sydney, Australia.

In the presence of memory constraints, it is mandatory to resolve multi-blockchain dependencies. This can be performed by moving toward network architectures with no managing authority. However, these architectures also suffer from other limitations such as service charging policies, record management, and tracking, and hence require further exploration.