BlockA*uth*: BlockChain based distributed producer Authentication in ICN

Mauro Conti*, Muhammad Hassan†, and Chhagan Lal§
Department of Mathematics, University of Padua, Padua, Italy *†§
Manipal University Jaipur, Jaipur, India§
Email: *conti@math.unipd.it, †hassan@math.unipd.it, §chhagan@math.unipd.it, †Corresponding Author

***Abstract*—In Information Centric Networking (ICN), consumer mobility is supported by design in virtue of its connectionless pull-based communication model. However, producer mobility management is challenging as it focuses on the named-based resolution mechanism, which applies a dynamic and direct interaction between the producer and forwarding plane. In this paper, we consider the fundamental security issues related to producer mobility in ICN. These security issues exist mainly due to the insecure interaction of producer with the network's forwarding information management system. We show that the current mobility solutions lack an adequate security mechanism and they invite severe security threats in the network (e.g., prefix hijacking and Denial of Service (DoS) attacks). To address such security threats, we propose a Blockchain based lightweight distributed mobile producer Authentication (BlockA*uth*) protocol to enable secure and efficient mobility management in ICN. BlockA*uth* authenticates the producers' prefix(es) and enforce them to express only genuine routing updates for the prefix(es) to which they are entitled to advertise. The qualitative security analysis confirms that BlockA*uth* is robust against various security attacks to which mobile network and blockchain are particularly vulnerable (e.g., prefix hijacking, double spending, DoS attack). Additionally, the performance evaluation of BlockA*uth* shows that it maintains significant performance gain compared to the state-of-the-art prefix attestation proposals. In particular, it maintains up to 94% of the network's original throughput, while it needs additional storage of just tens of megabytes.**

***Index Terms*—ICN, Mobility management, Authentication, Blockchain, Prefix hijacking, Security.**

I. Introduction

One major challenge in the Internet Protocol (IP) network is to provide efficient mobility management. Initially designed for the static wired network technology, IP failed to chase the technology evolution, which recognises now wireless connectivity and mobile devices. Different research efforts [1], [2], [3] have been carried out to overcome the lack of mobility support in the design of IP. However, none of them is fully able to provide a cost-effective and efficient mobility management mechanism.

ICN is an emerging networking paradigm, which meets the requirements of next-generation mobile networks (i.e., 5G) by adapting to multiple radio access technologies (e.g., Wifi and LTE). These requirements include global Internet access and seamless user mobility over dense and progressively heterogeneous network access. ICN offers native support for mobility at the network layer by decoupling the time and space among request resolution and content transfer. In particular, ICN replaces the host-centric communication (i.e., IP) approach to a content-centric approach. The communication in ICN is triggered by consumer entities who express interest for specific content. The network will then deliver the consumer's interest to the producer, which is entitled to generate the corresponding content under a specific name prefix (e.g., *Unipd/BMCS*), and later it forwards the content back towards the consumer.

Two fundamental characteristics of ICN architecture encourage seamless consumer mobility [4], [5]. Firstly, the communication model is receiver(consumer)-driven instead of producer in which the consumer uses the location-independent content names (in form of interest packets) to request the desired content. Secondly, the request/response communication model between consumer and producer is connection-less (i.e., *stateless*). Therefore, when a mobile consumer attaches to a new Point of Attachment (PoA), the above two characteristics permits the consumer to re-issue interests to obtain the data that it has not received from its previous PoA. However, the producer mobility is instead more challenging in ICN because of no separation between the routing locator and content identifier. Several proposals for handling producer mobility exists in the ICN literature [6], [4]. Among them, the *tracing(routing)-based* approaches tries to address the subject by updating the forwarding tables at each mobility event and then forwards the interest.

A. Motivation and Contribution

In tracing-based protocols [7], [8], [9], [10], [11], the producer is entitled to directly exploit ICN stateful forwarding plane to provide seamless mobility and to overcome the hand-off latency, packet loss, and signalling overhead. However, the tracing-based protocols also allows insecure interaction of producer with the network's forwarding(routing) information. Therefore, installing such protocols which are deprived of acceptable security mechanisms brings up serious security threats for all the network entities (e.g., consumer, producer and the network itself). In this regard, the producer should be allowed to issue only the legitimate routing updates, explicitly named as *Interest Updates (IUs)*, only for the prefix(es) it is entitled to publish the relevant content. In cases, where no adequate security mechanism exists to impose such rules, an adversary can easily forge IUs of the legitimate producers. Hence, the adversary can divert benign consumers request and network traffic towards itself, such an attack is known as

prefix hijacking [12] in ICN. By launching a prefix hijacking attack, an adversary is able to: (i) victimize benign users by performing blackhole attack [13], (ii) deny consumer access to their requested content [14], or make genuine content unreachable, and (iii) pollute the network caches with false content [15].

Currently, BlockChain (BC) is gaining significant attention from both academia and research industry where researchers exploit BC technologies to assure security, privacy, and access control for devices, data storage, and various other applications [16], [17], [18], [19], [20], [21], [22], [23]. Driven by the importance of addressing security issues at the initial stages of potentially new Internet architectures (such as ICN), we propose a BC based efficient & lightweight distributed mobile producer Authentication (BlockA*uth*) protocol for secure mobility management in ICN. BlockA*uth* authenticates the producers prefix(es) to enforce them to express only genuine IUs. To this end, the paper has the following major contributions.

- We propose a novel framework called BlockA*uth*, which uses blockchain technology to provide an efficient and lightweight technique to securely authenticate mobile producers in a distributed way. To the best of our knowledge, BlockA*uth* is the first framework that provides reliable, secure, and faster mobile producer authentication for ICN mobility management scenarios by using the blockchain technology.
- The proposed framework efficiently encounters the implicit necessities of ICN mobility management and cellular networks. It exploits the BC features such as time-based consensus algorithm, distributed trust, and throughput management, to enable a secure and efficient handover in ICN. Additionally, we propose a distinct global BC and expandable local immutable ledger designs to ensure seamless handovers in macro and micro network scenarios.
- We perform a qualitative security analysis of BlockA*uth* against various security attacks to which ICN mobile networks and BC are particularly vulnerable (e.g., prefix hijacking, double spending, and DoS). The results obtained through the performance evaluation of BlockA*uth* shows that it is able to maintain comparable performance to most of the existing hash-chain based prefix attestation proposals [24]. In particular, BlockA*uth* maintains 94% of the router's original throughput while requiring additional storage of just tens of megabyte to handle the authentication of millions of mobile producers.

B. Organization

The rest of the paper is organized as follows. Section II presents the overview of ICN and its application for next generation mobile networks (i.e., 5G). Section III presents an overview of mobility-enabling technologies in ICN and blockchain. Section IV discusses the fundamental security challenges and state-of-the-art with respect to ICN mobility management. Section V describes the design and working methodology of the proposed BlockA*uth* protocol, and Section VI illustrates scalable BC solution for BlockA*uth*. Section VII describe the security and performance analysis of BlockA*uth*. Finally, we conclude in Section VIII.

II. ICN Overview

At present ICN is gaining significant attention from both industry and academia due to its dynamic shift from host-centric to content-oriented paradigm [25], [26]. In particular, ICN implies the Publish-Subscribe Internet (PSI) architecture, which allows the user to primarily focus on the content instead of its retrieval location [27]. ICN has shown the potential to resolve several issues efficiently (e.g., mobility, security, and energy efficiency) that are ubiquitous in existing IP networks. These issues are particularly considered important in the next-generation telecommunication networks such as 5G [5]. Among many research efforts, Name Data Networking (NDN) [28] and Content Centric Networking (CCN) [29] are the two concrete variations of ICN which introduces a new network layer with the aim of replacing the existing TCP and IP. Both NDN and CCN are considered by the research community to be two reference projects implementing the ICN paradigm. Besides being prominent, the architectures of NDN and CCN are very similar, which make them almost indistinguishable for the scope of this paper. Therefore, to avoid any ambiguities, we refer to these reference architectures comprehensively as ICN in the rest of the article.

In ICN, the communication is accomplished through two specific type of packets, i.e., *Interest* and *Data* packets. Since the communication exploits the publish-subscribe model, the producer generates and publish the content using an explicit name prefix, e.g., *Unipd/BMCS*. The consumer can retrieve the content by issuing the named request called *Interest*. The ICN network implies the name-based routing fashion to forward the interests (e.g., *Unipd/BMCS/sb-forwarder.apk*) towards the producer to retrieve the matching content, i.e., data packet. An open-source implementation of NDN and CCNx gives more responsibilities to the routers as it introduces router-side content caching and interest aggregation [29]. Upon receipt of interest packet for a content, ICN router first checks whether a requested content is already present in the *cache* (i.e., Content Store). If the content is not found in the cache, the router looks in a *Pending Interest Table* (PIT) for a pending interest issued for the same content. The router forwards the interest towards its destination, in case, a PIT miss occurs. However, if there is a match in the PIT, further interests issued for the same name are not forwarded, instead collapsed in the PIT. Later, when the requested content arrives at the router, all the pending interests for it are satisfied. It is done just by sending the content back to all the hosts who issued those interests. In this way, ICN provides explicit support of multicast data routing. The router's *Forwarding Information Base* (FIB) is responsible for forwarding interests towards the content provider via one or more network interfaces (*faces*) based on the routes to the origin node(s). The requested data packet is then forwarded towards the sender by simply traversing, in reverse, the path of the preceding interest [28].

In contrast to IP, where security is provided by the upper layers to secure host-to-host communication, ICN comes with

security by design. In ICN, the content is explicitly shared along with the signature key of the producer. The key is used to verify the integrity of the received content. In particular, security in ICN follows a data-centric model. Therefore, the content is signed by the content provider to allow interest senders to verify its integrity and authenticate data-origin [30]. In this paper, we consider the security implications of producer mobility in ICN and we highlight the importance of securing producer to network interactions (later described in Section IV).

A. 5G-ICN

ICN is an emerging networking paradigm, which meets 5G requirements such as global Internet access and seamless user mobility over dense and progressively heterogeneous network access by adapting to multiple radio access technologies, e.g., Wifi and LTE. There are various advantages which can be achieved inherently through an ICN based 5G architecture (i.e., 5G-ICN) [31], [32], [33], [34], such as: (i) 5G-ICN provides a single protocol that is able to handle mobility and security instead of using a diverse set of IP-based 3GPP protocols (such as in the case of existing mobile networks, e.g., LTE, 3G, 4G), (ii) To connect devices and services to the network, it provides a unifying platform with the same layer 3 application programming interfaces (APIs) to integrate heterogeneous radios (e.g., Wifi, LTE, 3G) and wired interfaces, (iii) It converges services like computing, storage, and networking over a single platform, which enhances the flexibility of enabling virtualized service logic and caching functions anywhere in the network. Studies in [32], [34] show that 5G-ICN is capable of enabling a flat architecture for the data plane, i.e., without any specialized gateways. As shown in Figure 1, application devices are connected through Radio Access Network (RAN) to ICN gateways, thus also acting as service enabled RAN (SE-RAN). In Figure 1, edge ICN routers also function as ICN service routers (ICN-SRs) since these routers are provided with additional computing and storage functionalities. Similarly, ICN support to IoT applications enables efficient delivery of IoT services using the same protocol infrastructure [35]. The inherent ICN features (e.g., in-network caching and computing, and multi-homing) enables 5G-ICN to offer support for high bandwidth applications naturally. Furthermore, there are several requirements which 5G-ICN meets and are not addressed by the existing cellular network architectures (e.g., LTE/4G), and these include:

- Naming: The applications in ICN are bind to unique names which are used to identify hosts, contents, or services. Thus, naming the resources protects applications from any host or service mobility. It is because the ICN layer handles the mapping of higher-layer application identifiers to network identifiers.
- Mobility: ICN enables a flat architecture, where mobility is managed in a distributed manner by the point of attachment nodes, later explained comprehensively in Section III-A. As shown in Figure 1, the ICN base stations (ICN-BS) or ICN-SRs, which are integrating multiple radio access networks are responsible for doing that. In contrast, mobility in LTE/4G is managed through a complex set of orthogonal protocols.

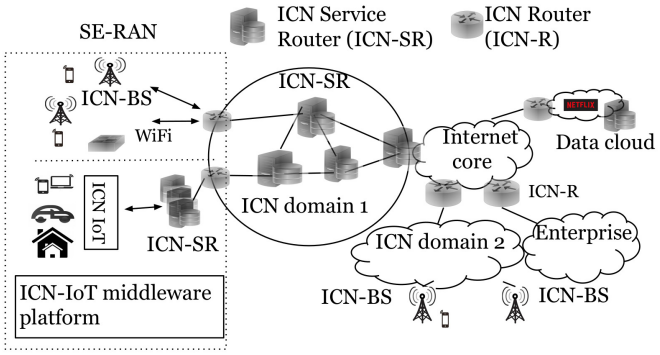

Fig. 1. ICN enabled 5G architecture [32]

III. Background

In this section, we illustrate the overview of mobility management technologies in ICN and their functionality. Later, we present an overview of BC technology.

A. Mobility management in ICN

In contrast to IP networks where handling mobility requires cumbersome solutions such as Mobile IP [1], [3], ICN provides native mobility support to consumers due to its following two fundamental characteristics: (i) the communication model is receiver(consumer)-driven, where a consumer uses the location-independent content names to request data, while in the current Internet architecture sender has complete control on data transfer, and (ii) the request/response communication model of ICN between consumer and producer is connectionless (i.e., *stateless*). It is in contrast to current TCP/IP connection-oriented (*stateful*) end-to-end communication, which requires a binding between user's location and address. Therefore, when mobile consumer attaches to a new PoA, the two characteristics mentioned above permits the consumer to reissue the interests to obtain the data, which he/she did not receive at its previous PoA. In this way, consumer achieves seamless mobility support in ICN, deprived of rebuilding a TCP connection or through cumbersome and overwhelm IP mobility patches [1]. However, producer mobility is more challenging in ICN because of no separation between the routing locator and content identifier. In particular, for each mobility event initiated by a producer, the network should maintain producer reachability, and the routing devices must adjust their forwarding information so that the interest(s) matching the prefix(es) (i.e., owned by the producer) can be redirected to its new location. Thus, unlike consumer mobility, ICN requires updating of name resolution system over the new location of the producer to maintain routing consistency [4], [5].

In the past, few proposals for handling producer mobility are proposed [6], [4]. The solutions like *indirection-based* and *resolution-based* supports producer mobility. However, these also bring complexities to few fundamental problems such

as handoff latency and packet overhead during encapsulation and decapsulation that leads to QoS degradation [9]. The *routing(tracing)-based* approaches try to address the subject by updating the forwarding table at each mobility event. In particular, the tracing-based protocols [7], [8], [9], [10], [11] directly exploits the ICN stateful forwarding plane to overcome the handoff latency, packet loss, and signaling overhead. However, the tracing-based protocols allow the producer to interact with the network forwarding information directly. Hence, installing such protocols deprived of acceptable security mechanisms could cause serious security threats for all the ICN entities.

B. Blockchain

BC [36] is an immutable time-stamp ledger of hashed blocks which functions to store and share data in a distributed manner [37]. In recent years, practitioner and academics in diverse disciplines (e.g., law, finance, and computer science) have been tremendously attracted by BC due to its noticeable features such as distributed structure, immutability, security and pseudo-anonymity [18], [38], [17], [16]. In BC, each data block is hashed and linked with the previous block to provide immutability. A data block contains a number of verified instances (called transactions) and some required control information (called header). A chain of blocks can be replicated and spread to all participants in the BC network such that the data contained in the BC is synchronized globally. Figure 2 illustrates the basic structure of BC.

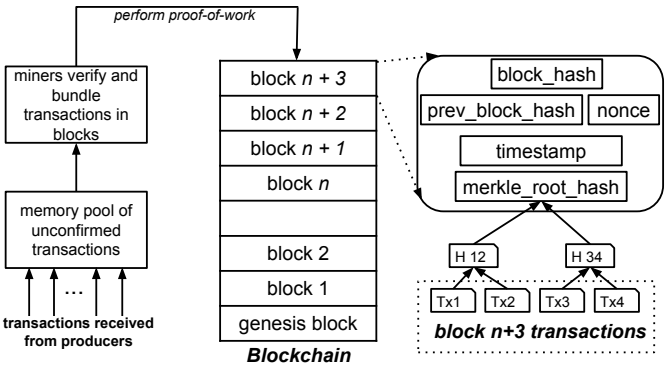

Fig. 2. A generic Blockchain structure

Each new transaction is verified and confirmed by all the participating nodes in the network. Therefore, it eliminates the necessity for any central authority. Appending a new block to the BC (referred to as a mining process in literature) may entail solving a computationally demanding, hard-to-solve, and easy-to-verify puzzle. The puzzle indeed supports a trust-full consensus algorithm among the untrusted participating nodes. The example of typical consensus algorithms used in blockchain implementation are Proof of Work (PoW) [39] and Proof of Stake (PoS) [37]. To propagate the transactions and blocks to update the ledger, BC operates with multi-hop broadcast functionality. BC [36] technology was first devoted to power the bitcoin cryptocurrency, but nowadays it is progressively proving its applicability to various other applications. For instance, authors in [40] utilizes BC to guarantee privacy aware and secure personal data management. In [41] authors describe decentralized, secure content access using BC in ICN based platform. In the proposed protocol, we utilize BC to store the access control data in a decentralized manner to authenticate mobile producer interaction with network forwarding information. Also, with the use of BC we assure trust and security in the network in a distributed manner.

IV. Related Work

In the literature of IP based mobile networks, the prefix hijacking attack is mitigated by adopting prefix attestation mechanisms in IP-based mobility protocols such as Mobile IP [1], Cellular IP [42], and TeleMIP [43]. In all these approaches, the host has been assigned with a host ID and session key by the network gateway during its initial attachment to the gateway. In the case of handover, the host uses the session key for IP address authentication owned by it to the new PoA. However, the approach brings some limitations when it comes to ICN-based future mobile networks, i.e., 5G-ICN. Since the protocols mentioned above follow a centralized authentication mechanism, therefore, whenever the host changes PoA, a central entity is needed to authenticate the host [1]. It is worth noticing that the efficacy of ICN tracing-based mobility protocols relies on the elimination of any central entity (i.e., managing the mobility of producers) to overcome handoff latency. Moreover, in approaches such as [42], [43], the use of a single network key to generate host's session keys is also problematic. For instance, if the network key is compromised, then any router can be compromised, and new network keys along with the regeneration of all sessions keys are needed to tackle the issue. Also, these mechanisms do not provide any solution to identify malicious network routers and legitimate malicious users, which tries to perform repudiation or replay attacks.

Some authors also addressed the issue of IP prefix hijacking in inter-domain [44], [45] and intra-domain [46], [47] IP routing. The authors propose prefix attestation to mitigate the threat. One common mechanism to achieve IP prefix attestation is by exploiting digital signatures and certificates, in which an address owner requests a signed certificate to attest the routers right to announce specific IP addresses in the network. For instance, sBGP [48] and soBGP [49] make use of public key infrastructure to maintain trust between the network and the address owners. Correspondingly, authors in [46] propose the use of signed certificates for the network prefixes which OSPF routers need to announce in different OSPF areas. Similar approaches can be applied to the ICN tracing-based mobility protocols. However, it would suffer from similar issues, e.g., centralized authentication management, repudiation or replay attacks.

Recently, authors in [24] highlight the issue of prefix hijacking [12] in ICN. To address it, the authors propose a distributed prefix authentication protocol for producer mobility in ICN tracing-based protocols. The protocol utilizes a one-way hash-chain mechanism to guarantee that a producer can only generate legitimate IUs for its prefix(es). Moreover, to achieve the forward secrecy, the network and the producers are

always forced to maintain a synchronized state of the current hash chain value, which is used for prefix authentication. We identify that similar to other approaches, the protocol in [24] also do not completely prevent the prefix hijacking and several other attacks such as DoS, replay and double spending. For instance, there is no guarantee that every router in the network will have the most recent version of the forwarding state, i.e., a synchronized sequence number of the current hash chain value. Hence, the old routers could not be able to detect old or replayed IUs if the IUs holds a greater sequence number than the security context stored at the router. Secondly, the one-way hash computation also encourages DoS attacks to the edge routers. For instance, adversary issues a non-legitimate IU holding a sequence number (say j), such that $j >>> i$, where i is the recent sequence number that has been used at the edge router. As a result, to detect the non-legitimate IU, the router is forced to hash $j - i$ times the security context associated with the prefix. The greater is the distance between j and i, the more the number of hashes router have to compute. An attacker can issue non-legitimate IUs with great sequence numbers to keep the router busy on calculating such hashes. Thus it provokes a DoS attack to the other connected producers. Additionally, the protocol also lacks to mitigate the double spending attack, e.g., repudiation by a legitimate producer. Categorically, it is the case where a legitimate producer tries to use the same sequence number for a hash value (nearly at the same time) on two different edge routers in the network. The producer can re-route legitimate traffic to or for that producer on different paths since the hash chain is not able to identify the double spending of sequence number by the producer. Finally, the protocol also lacks to provide the mechanism to identify malicious routers and producers which are connected to the network.

V. BlockA*uth*: BlockChain based Distributed Producer Authentication

In this section, we present the design and working methodology of the proposed BC based efficient & lightweight distributed mobile producer Authentication (BlockA*uth*) framework, which offers a secure and fast mobile user authentication. It also mitigates various security issues of mobility management in ICN.

A. BlockAuth Architecture Framework

To perform efficient handover in inter and intra-clusters [50], [51], [52], [53], the BlockA*uth* architecture framework consist of two main tiers namely: core network, and clusters[1] (i.e., micro-cells). Each cluster consists of a group of access gateways (e.g., ICN base stations, WiFi access points) and one cluster head, as illustrated in Figure 3. In particular, some of the ICN routers which entails the necessary functionalities such as processing capability and bandwidth availability are selected as Cluster-Head (CH). The base stations register with the nearest CH to become the member of the respective cluster. BlockA*uth* utilizes the *weighted clustering algorithm (WCA)* [54] which takes into consideration the number of base stations that a cluster-head (i.e., edge router of the core network) can handle efficiently without any severe degradation of the system performance. In particular, while selecting CHs, WCA considers the transmission power, mobility, and battery power of the mobile nodes.

[1]The concept of dividing the geographical region into small zones has been presented essentially in the literature as clustering.

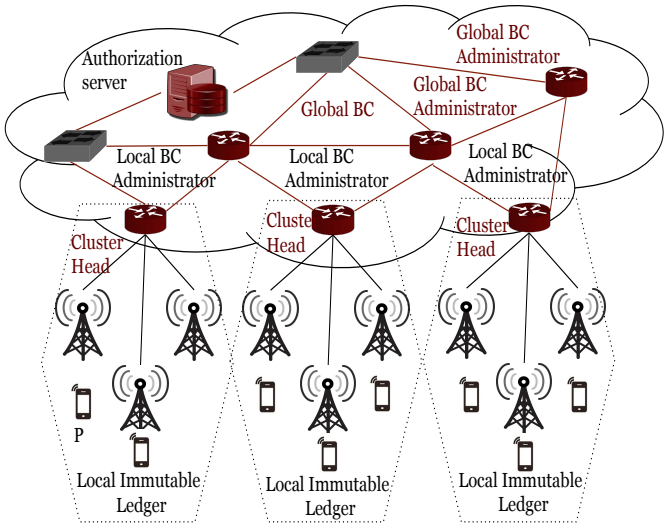

Fig. 3. System model of BlockAuth framework

To ensure fast and seamless intra (also named as micro) cell handover to reduce signaling overhead, the base station (or access point) within a cluster uses a private expandable immutable ledger entailing the transactions called as local transactions. Categorically, the structure of private immutable ledger is similar to BC. However, it is managed solely and centrally by the respective CH, later we name it as Local Immutable Ledger (LIL). To guarantee the requirements of handover latency and scalability, only the CH is responsible for managing private LIL. The CH is thus also named as Local Immutable ledger Administrator (LIA).

In BlockA*uth*, the capable routers in the core network collaboratively manage the permissioned Global BC (GBC), which stores the transactions generated to and from various clusters to perform inter-cluster (or macro) handover [52]. In particular, the BC in the core network is managed by a subset of the core routers which we call as Global BC Administrators (GBA). Thus, the routers participating in GBC (including CHs) are responsible for processing the incoming and outgoing transactions that are generated by mobile producers from different clusters. Table I presents the summary of notations used in the paper.

B. System And Adversary Model

In this paper, we consider the scenario of anchor-less forwarding [7], [55] for mobility management. The proposed model follows the scenario of handover mechanism between base stations similar to [56], [57], and it includes macro and micro mobility scenarios, as illustrated in Figure 3. However, it does not specialize to any specific wireless media, i.e., the access gateways can be LTE/4G/5G cell or WiFi access points.

TABLE I
SUMMARY OF NOTATIONS

Notation	Meaning
As	Authorization server
BA	Blockchain Administrator
GBA	Global BC Administrator
LIA	Local Immutable ledger Administrator
$\|R\|$	set of routers
R_i	$R_i \epsilon \|R\|$
Bs	base station/access gateway
CH	Cluster Head
P	mobile Producer
Adv	Adversary
IU	Interest Update message
$prefix$	producer prefix
$H(.)$	cryptographic hash function
pk^s_P, sk^s_P	public and private key pair associated with P
$enc(.,.), dec(.,.)$	public key encryption and decryption function
Tx^{ID}	Transaction ID
$previous\_Tx^{ID}$	previous Transaction ID
Tx^{Max}	maximum Transactions in a block
Tx_i^{ID}	local IL Transaction ID
Tx_j^{ID}	global BC Transaction ID

The significant entities involved in the proposed protocol are Authorization Server (As), core routers and CHs. The CHs are responsible for managing their respective cluster members and the mobile producers attached to them. The subset of core routers functioning as GBAs manages the GBC, and if the GBA is CH than it leads the LIL as well. Furthermore, we consider that the composition of the whole network containing the edge and core routers forms a single autonomous system. The mobile devices are aware of their valid credentials to connect to the proper network infrastructure, i.e., producers are provided with valid Subscriber Identify Module (SIM) devices by the network operator. Moreover, once the authentication is performed, the communication between each mobile device and access gateway is considered secure.

1) System model: To accomplish the verification of IUs that are issued by a mobile producer (P), we explicitly categorize P as a mobile device. The device stores $P's$ identity, which is entitled to publish content using one or multiple $prefix$(es). Each $prefix$ (i.e., owned by P) is associated with a pair of public/private key, say pk^s_P and sk^s_P respectively, which are used to sign/verify the $prefix$(es) that P publishes. We assume an additional field attached to IU along with the prefix, which entails the security context to verify the IU.

In the protocol, authentication server (As) is mainly responsible for performing the:

- first authentication of P and verification of the $prefix$(es) owned by P,
- generation of the genesis transaction for BC which serves as a starting point of the global BC.

2) Adversary Model: We consider an adversarial model in which an adversary (Adv) is capable of controlling mobile devices that can attach to the network, e.g., an attacker owns valid SIM cards and connects to the network. The Adv could target the authorized mobile producers and purposely generate legitimate IUs for the $prefix$(es) used/owned by its victims. Also, a legitimate P can also be Adv, and it aims to corrupt the network by double-spending attacks. For instance, P can try to use the valid security context for the $prefix$ authentication on two or more different base stations (Bs) at the same time. We also assume that Bs, CH, and core routers can be compromised by the Adv, while the As is considered a trusted entity. These assumptions are consistent with the assumptions made on the existing heterogeneous mobile networks [58]. Moreover, we assume there is no intrusion detection mechanism in place. Finally, we also assume that Adv can access information stored at the compromised ICN router (R_i), including Bs, LIA, and GBA nodes.

C. Modelling Data through BlockChain

As previously illustrated, BC functions as a transaction database which is distributed and shared among all nodes participating in the BC network. We exploited BC as an application for distributed data storage which provides various functionalities for data storage [19], [59]. Among them, two basic primitives that are essential to the proposed solution are: (i) retrieve transaction, and (ii) add transaction. In this section, we illustrate how BC provisions these primitives and enables the data flow in the proposed model. However, first, we review some key definitions of BC characteristics in the light of BC adopted by Nakamoto [60], as illustrated in Figure 2.

- Transaction: We express each $P's$ $prefix$ authentication request (i.e., IU) as a single transaction unit in BC.
- Block: Multiple valid transactions are clubbed together to form a block, and the block is verified before storing into the blockchain. Also, BC stores the blocks linearly in a chronological manner over time. Same as the transactions, each block also delivers immutability by containing the hash of the previous block.
- Mining: During mining, all the miners validates each transaction, create blocks, and then verify the blocks to add them in the blockchain. In BlockA*uth*, the process of mining is only performed by BC administrators, i.e., GBAs and LIAs. Therefore, only a subset of core routers and CHs are responsible for mine and broadcast the new blocks into the network.
- Genesis block: It is the first block in the blockchain. In BlockA*uth*, the As is responsible for generating the genesis block for the BC instantiation.

BlockA*uth* exploits BC as a transaction database shared among all routers, which are responsible to authenticate $prefix$(es), i.e., IU published by P. The global BC consists of transactions added in sequential order, referring back to the very first one (i.e., when P first register to the network). The entire BC is private among network nodes, and a non-member can not openly review it. The two essential primitives enabled by blockchain are as follow.

- Retrieve transaction: As per the design, a local copy of global BC is available on all the ICN routers. The BC data can be retrieved by any router (say R_i) to perform $prefix$ authentication. Each new transaction received by an R_i is authenticated by using the relevant information of producer that is stored in BC. In particular, when a new $prefix$ authentication request (i.e., new transaction generated by a producer) is received by an R_i, it retrieves

the previous transaction of the producer by using the previous transaction ID ($previous\_Tx^{ID}$) value given in the received transaction. By retrieving the previous transactions of a producer from BC, the routers are able to retrieve all the relevant security context used previously for the same $prefix$ authentication [20], [19].

- Add transaction: Once a transaction is authenticated by using the process mentioned above in the retrieve transaction phase, it will be added in the blockchain. In blockchain, instead of adding individual transactions, the miners wait to collect a predefined number of new transactions. When there are enough transactions in the mining pool these are bundled to create the next block which will be mined and added in the blockchain. During the mining process, all the miners start competing to generate the next block (later, described in Section VI-A1). The miner who mines a new block first will append it in its blockchain, and it also broadcast the same to other R_i in the network [39], [37]. When a newly mined block is received by an R_i, it verify and add it to its local copy of BC. In this way, each R_i will have a the same (i.e., most updated) copy of the global BC.

D. Initial producer authentication

In BlockA*uth*, mobile producer (P) entails the pk_P^s and sk_P^s key pair, which is associated with the $prefix$(es) owned by P. It is accomplished similar to the traditional mobile networks, i.e., by utilizing the mobile device's SIM card which is provided to P by the network service providers [56], [57]. The sk_P^s key of P is hard-coded on SIM card, and it is securely distributed and registered with its respective pk_P^s key at As. When P connects with the network first time, the As directly verifies the $prefix$ announced by P. In particular, a P issues a registration interest with the $prefix$ and pk_P^s. The interest (aka transaction) along with $prefix$ also carries a digital signature that P generates by encrypting $prefix$ and some additional information and then sign it with its sk_P^s key. The additional information consists of the IDs of Bs and CH to which producer is currently attached, and the IDs related to BC specific fields (i.e., Tx^{ID} and $previous\_Tx^{ID}$). Recall that Tx^{ID} is required to link a Tx to its previous Tx in BC. In particular, Tx^{ID} is the hash pointer of the whole interest message, and it is calculated using a pre-defined hashing algorithm, i.e., $H(.)$, such as Tx^{ID} is the identifier of the current transaction message, where $Tx = H(message)$.

When the registration interest (i.e., first interest generated by P to advertise its $prefix(es)$) is sent by the producer, it holds a null value for $previous\_Tx^{ID}$ field. An ICN node receiving such an interest, forwards the interest towards As. Once the As receive the interest, it verifies the $prefix$ by validating the digital signature of P. After successful verification, As generates the first transaction (i.e., the first transaction for a particular $prefix$ of P in the BC) for P in the BC and broadcast it in the network. All the miners that receive the transaction adds it in their next block, which they form to mine. The first transaction of P includes the following data: (i) Tx^{ID}, which also becomes the output of this specific transaction, (ii) $previous\_Tx^{ID}$, (iii) pk_P^s key assigned to P, (iv) digital signature sent by P to As for prefix verification, and (v) payload, which includes information to increase scalability and efficiency of BC (we later describe this in Section VI-A3). Figure 4 illustrates the steps taken to perform initial registration of P in the protocol.

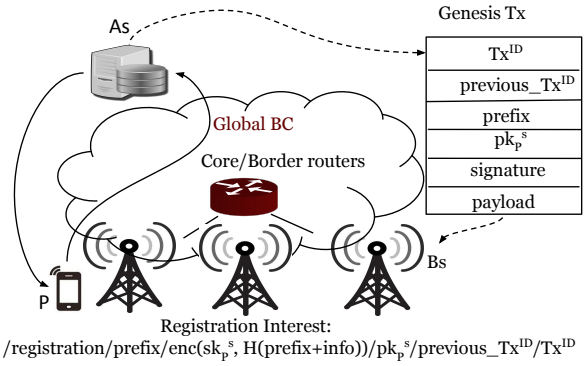

Fig. 4. Initial registration and authentication process of producer

E. Secure producer mobility

In this section, we present the details of IU authentication mechanism, i.e., when a mobile P attaches to a new PoA (i.e., Bs) and issues a new IU following the taxonomy of [7], [55]. Similar to Section V-D, in the IU authentication mechanism, P needs to send IU which comprises of $prefix$ and security content signed with the its sk_P^s along with Tx^{ID} and $previous\_Tx^{ID}$.

In BlockA*uth*, we use the Public Key Infrastructure (PKI)-based infrastructure along with BC technology for prefix attestation. The idea of Pretty Good Privacy (PGP) encryption program uses PKI to provide the following three main functionalities (i) confidentiality with encryption, (ii) authentication via digital signatures, and (iii) web of trust via identity validation from peers. We exploit the PKI [61] as a significant function to perform $P's$ $prefix$ authentication in mobile networks using BC technology. In particular, BlockA*uth* utilizes BC as a database to store pk_P^s, digital signature, and additional information of P to allow each router in the network to authenticate IUs generated by P. Besides, it also validates the integrity of ICN nodes (i.e., R_i) that participate in the BC network.

To securely verify the IU during handover, the producer sign the $prefix$ along with the currently attached Bs and CH IDs with its sk_P^s. Along with it, the $prefix$, pk_P^s, the current Tx^{ID}, and the previously used transaction identifier, i.e., $previous\_Tx^{ID}$, is also sent as plain text. In particular, at each instance when P connects to a new Bs, it issues an IU consisting of the following fields.

- $prefix$[2].
- Public key (pk_P^s) associated with P's $prefix$, which is required in the network for the signature verification of IU.

[2]Note that a plain text name prefix is needed to route the interest towards previous PoA and update the forwarding information.

- Digital signature, which consist of $prefix$ and additional information signed by the private key (sk^s_P) of P, i.e., $enc(sk^s_P, H(prefix+info))$. The signature ensures $P's$ $prefix$ authenticity and immutability with it's previous authentication event.
- Previous BC transaction identifier ($previous\_Tx^{ID}$), which is used to find previously stored authentication data of any specific P in BC.
- Current BC transaction identifier (Tx^{ID}), it is the hash of the whole IU, i.e., $Tx_{ID} = H(IU)$.

The purpose of signing the Bs and CH IDs along with the $prefix$ is to make the signature immutable with the initial authentic $prefix$ registration, and it also helps to keep track of all previous mobility activities of P which could be used for analysis purposes. In particular, BlockA*uth* aims to achieve the *backward secrecy* by linking each IU authentication of P with its previous authentication event in an immutable way through BC, and the process will eventually lead towards the initial $prefix$ registration of P. In particular, on each IU authentication event, a new value of Bs or CH ID[3] is signed along with prefix. However, the $prefix$ value remains the same. Thus, to generate a different digital signature, the BlockA*uth* utilizes Bs or CH ID as a piece of additional information along with $prefix$. Additionally, the process ensures that current IU message for authentication is received from the same P which has earlier performed a successful IU authentication. It is because the previous signature composed of previous additional information (i.e., previous Bs/CH IDs) is stored in the BC and immutability is linked with the current transaction through $previous\_Tx^{ID}$.

In our illustration, we only show the authentication steps performed at Bs. However, it has to be considered that each R_i participating in the network (i.e., core and border routers) execute the same authentication steps upon reception of a new IU request. The Bs while receiving the IU, first retrieve the previous transaction of P by using $previous\_Tx^{ID}$, which is available in current transaction. The $previous\_Tx^{ID}$ is a hash pointer towards $P's$ previous transaction in BC. To authenticate the IU, the Bs fetches the pk^s_P from $previous\_Tx^{ID}$ transaction and uses it verify the digital signature stored in current transaction. If the digital signature is verified correctly, then the Bs authenticates the P and forward the transaction to CH. The CH broadcast the transaction in the network so that it can be added in the BC. Figure 5 illustrates the message flow for initial registration and IU authentication mechanism of BlockA*uth* protocol in detail.

VI. Efficient & Scalable BC for BlockA*uth*

In this section, we discuss the mechanism of BC transaction generation and verification in the core network and individual clusters (also referred to as global BC and local IL transactions). To exemplify the functioning of BC in two fundamental tiers, we commence the discussion by explaining the following regimes.

[3]Since on each handover Bs or CH ID changes, therefore it triggers further additional information.

- *Transaction:* It is symbolized as a general communication primitive, which is utilized to exchange authentication control information for BlockA*uth*. Recall that as stated prior, the network data flow in BlockA*uth* is separate from transactions.
- *Global BC Administrator (GBA):* It is an entity which is responsible for managing the global BC. The critical operational task of BA is to verify, broadcast, and store all the received valid transactions. The role of BA functioning in a core network and clusters consist of slightly different actions which are explained below in detail.

A. Global BC Transaction Process

The core network potentially consists of multiple core and edge routers. To ensure scalability, we assume that the subset of core routers and all CHs are managing the global BC, and these routers are named as Global Blockchain Administrators (GBA). Besides, each CH is also functioning as a Local Immutable ledger Administrator (LIA). The functioning of LIA is illustrated in detail in Section VI-B. The CHs process and manage all transactions that are coming to and from their respective cluster members.

Similar to Bitcoin [60], to ensure the integrity of GBAs, the blocks generated by GBAs are secured through asymmetric encryption, digital signatures, and cryptographic hash functions (e.g., SHA256). In contrast, for a transaction to be considered valid, the protocol requires only a single signature transaction, which is the signature of the requester, i.e., P. The structure of a transaction in BlockA*uth* is shown in Figure 6. The first field in the structure is an identifier for the current transaction, which is the hash of the entire IU message. The second field is a hash pointer towards the previous transaction of the same P. In this way, all the transactions generated by each P are chained together, and it is followed by the original prefix, public key and digital signature (i.e., $prefix$ and additional information signed by the private key) of the mobile P. Further additional data is also stored in the fifth field of the payload of Tx, which could be used for various purposes. In BlockA*uth*, *Distributed Trust Algorithm* use this specific data to optimize the efficiency of BC [18] while updating related communication messages. The algorithm is detailed below in Section VI-A3.

All the transactions in the core network follows the genesis transaction, which is generated during the first instantiation of the global BC by the As. As the transactions are stored in the blockchain in terms of blocks, each block comprises of two core portions: a set of authenticated transactions and a block header. The header of the block contains the hash of the previous block, block generator (i.e., GBA) ID, and signature of the block generator. The hash of the preceding block in the BC safeguards immutability at block level. If an adversary attempts to corrupt any previously stored transaction in any of the blocks that are stored in the BC, then the hash of the subsequent blocks that are stored on top of it will no longer be consistent with the global BC. Hence, it will expose the attack. Similar to Bitcoin, multiple transactions are grouped together and then processed as one block. A block can store

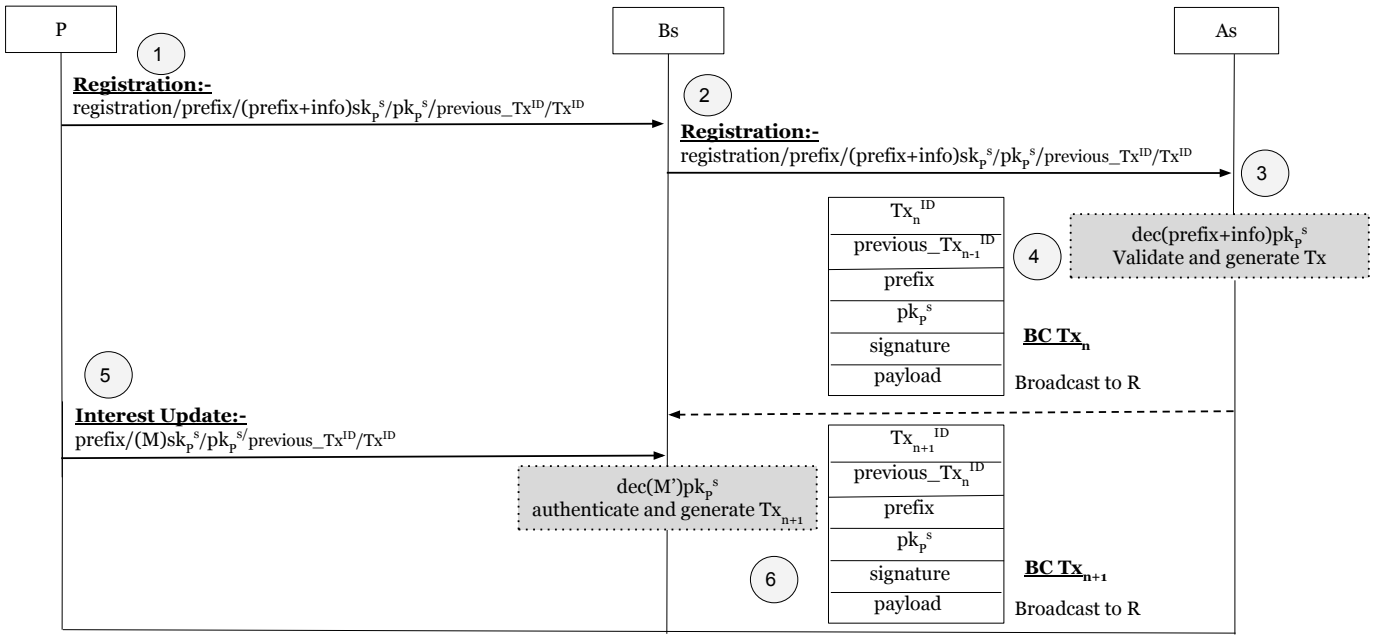

Fig. 5. Message flow for BlockA*uth*

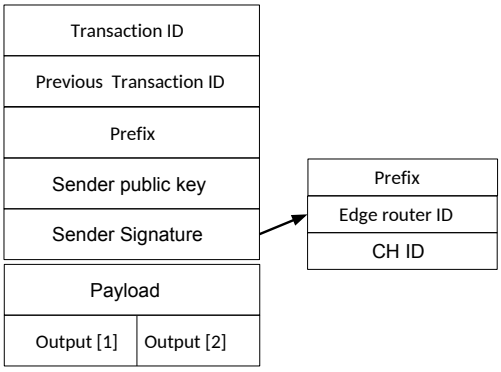

Fig. 6. Transaction structure that a producer uses for (re-)authentication in BlockAuth

at most Tx^{Max} transactions. The value of Tx^{Max} relates with the BC throughput.

When the CH receive a transaction (say x), it first check whether the P that generated the transaction has moved within the same cluster or not. It is achieved by forcing the P to issue two distinct pair of transaction IDs, one related to local Immutable Ledger (IL) (Tx_i^{ID}) and other to global BC (Tx_j^{ID}). The CH (which is also a GBA) will first perform a look-up for Tx_i^{ID} in local IL. In case, the local IL transaction ID has no pointer found for the previous transaction, then the CH uses the global transaction ID to retrieve the previous immutable transaction generated by the P in global BC. In contrast, if the incoming transaction matches the hash pointer for Tx_i^{ID} in local IL, then CH processes the transaction and updates the local IL among the cluster members. The procedure of managing the local IL is later detailed in section VI-B.

If a transaction is generated when the P moves from one cluster to another, then the transaction is verified using Tx_j^{ID} by the new CH and also broadcasted to all other GBAs. All global transactions are being verified by each router and are stored in a local, unprocessed transaction pool at each GBA. When the size of the transaction pool reaches to Tx^{max} then the GBA creates a block using the transactions in the pool, and it starts the validation process which is followed by the block inclusion in the global BC utilizing a consensus algorithm.

1) Consensus algorithm: Instead of using traditional resource-intensive consensus algorithms such as PoW or PoS, BlockA*uth* exploits a time-based consensus algorithm given in [18], [21]. The consensus algorithm ensures that a block generator is nominated randomly among all miner nodes (i.e., BC administrators) that are participating in the protocol. Moreover, the block generator is limited to the number of blocks a node can generate within a particular duration. To enforce randomness among block generation to avoid forging, before generating a new block every single BA administrator is forced to wait for a random time known as *waiting-period*. Due to different waiting-period experienced by each BC administrator, an administrator might receive a new block created by another administrator which contains some or all of the transactions currently present in its pool of transactions. Therefore, in this case, the BC administrator receiving the new block will remove all those transactions from the pool that has been already stored in the BC by the recently added block(s).

By forcing the block generators to wait for a random time reduces the duplication of blocks, which can be generated simultaneously. The maximum waiting-time is capped at twice the maximum end-to-end delay between the routers in the core network. The cap is used to ensure that there is sufficient time for disseminating a newly generated block by other BC administrators. When a new block is generated, it is

broadcasted to all other routers so that it can be appended in their local copy of global BC.

To protect the overlay against a malicious BC administrator which can potentially generate a large number of blocks with fake transactions that leads to an appending attack (later discussed in Section VII-A), the periodicity with which an administrator can generate new blocks is restricted such that only one block can be generated over an interval denoted by the consensus-period. Any non-compliant blocks are discarded, and the trust associated with the responsible BC administrator is decreased as outlined in the Section VI-A3. The consensus-period is adjusted by *Distributed Throughput Management* (DTM) which we discuss in Section VI-A4.

2) Verification: Every BC administrator validates each receiving block before appending it to its local BC. To validate the block, the signature of the block generator is validated. It is assumed that each BC administrator uses a predefined asymmetric keys[4] for block generation and communication. Each transaction in the block is also verified, therefore, similar to Bitcoin, a block is considered to be valid only if all transactions in the block are valid.

Algorithm 1 outlines the procedure for verifying an individual transaction (say x). As discussed in Section V-E, the link between the successive transactions of a P is established by including the hash pointer of its previous transaction. Thus, the BA first confirms this linkage between consecutive transactions of a P by comparing the hash of its previous Tx ID in x with the first field of the previous transaction (i.e., $x-1$). Following this, the digital signature of P stored in the fourth field of x is verified using its pk_P^s from $x-1$. Although, the pk_P^s is available in x, but it is retrieved from $x-1$ to ensure that the pk_P^s is linked with the first transaction of the P, in which pk_P^s is verified by the As.

Algorithm 1 BlockChain Tx verification

Input: $Tx\_verification\_proc$(prefix, pk_P^s, M', Tx^{ID}, $previous\_Tx^{ID}$)
Output: *True or False*

```
if previous_Tx^ID ≠ Tx − 1^ID then
    return False
else
    if enc(sk_P^x, M) ≠ dec(pk_P^{x−1}, M') then
        return False
    else
        check prefix with M'
        return True
    end if
end if
```

3) Trust association among BC administrators: For the network routers participating in BC formation, it is computationally challenging to verify all transactions and blocks. Particularly, when the number of macro-mobility events (i.e., inter-cluster) in the network increases. In BlockA*uth*, routers do not need to verify complete BC instantiation at each instance. To preserve the required immutability and to ensure scalability considering smooth handover requirements, we exploit *Distributed Trust Algorithm* (DTA), which gradually reduces the number of transactions needed to be verified by BC administrator in each newly mined block, before the block is added in its local copy of blockchain. It is achieved by building a trust relationship between the routers generating the new blocks, i.e., GBAs. The trust algorithm builds the concept of primary and secondary evidence between GBAs as follows:

- Primary evidence: The evidence in which a GBA (say A) has at least one previously verified block generated by another administrator (say B) is called as primary evidence of B for A.
- Secondary evidence: If GBA A does not have any primary evidence about B, however, anyone among other GBAs has established that the block generated by B is valid, then A has a piece of secondary evidence about B.

All GBAs maintain a list, which stores the relevant information regarding primary and secondary evidence. For instance, the administrator records the list of blocks and senders that are verified. Therefore, considering the attacks in which an administrator might generate the blocks that are not compliant with the proposed consensus algorithm results in decreased trust association. Hence, for one discarded altered block, the receiver decreases the trust association by a factor of one for the malicious sender. If the malicious BC administrator remains with this conduct, it faces with consistently reduced trust rating, which results in more and more of its transactions being verified before accepted for further processing by the BCAs. Vice versa, for one accepted verified block, the receiving GBA increase the trust association of sender by one. In the case of secondary evidence, the administrators check the number of other administrators that have verified the received block to get the trust association of block generator. The key benefit of the trust algorithm is that stronger the evidence is received for an administrator generating blocks, lower the number of transactions is the blocks generated by it that need to be verified before adding into the BC.

In Figure 6, the output [0] and output [1] values given in the payload are used to create the reputation of GBA that is sending the transaction (x). Initially, the sending GBA set these outputs based on its history of blocks. If the receiving GBA accept a block, then it will increase the output [0] by one. Otherwise, it increases the output [1]. To protect the BC against routers that claim false reputation by increasing their outputs before sending them to the other GBAs, during block verification, the receiving GBA checks that only one of $x's$ outputs, i.e., either the number of successful blocks (i.e., output [0]) or the number of rejected blocks (i.e., output [1]) is increased by one.

Aiming at the performance and objectives of BlockA*uth*, Figure 7 elaborates the functionality of trust association algorithm, which a GBA follows while verifying the transactions of a block received by another GBA. In particular, the primary evidence takes precedence over the secondary evidence. For

[4]We propose a solution implementing permissioned BC. Therefore BC designer, i.e., network owner, is responsible for key distribution in the core network.

instance, if a GBA has the primary evidence about any block generator, e.g., 50 earlier received blocks by a GBA are verified, then only fraction of transactions within the current block are selected for verification[5] (refer to Figure 7). Correspondingly, in the case where no primary evidence is available for the block generator, the GBA first checks for the availability of secondary evidence. The secondary evidence indicates the percentage of other GBAs that have vouched for the block generator in question, e.g., in case only 20% of other GBAs have asserted the block generator then 80% of the transactions within the received block needs to be verified (refer to Figure 7). Finally, if no evidence is recorded, then all the transactions in the block are verified. Note that a certain fraction of transactions are always required to be validated even if there is a piece of strong evidence, it is to protect against any potential malicious BCA.

Primary evidence	Previously validated blocks	10	20	30	40	50
	Transactions Required to validate	80 %	60 %	40 %	20 %	10 %
Secondary evidence	% of BAs signed the block	20 %	40 %	60 %	80 %	100 %
	Required to validate	80 %	60 %	50 %	40 %	30 %

Fig. 7. Trust Association among BA

4) Distributed throughput management: To strengthen the throughput performance of the proposed BC solution, we make use of a *Distributed Throughput Management* (DTM) mechanism [18], [21]. The BC throughput is measured with the number of transactions added in BC per second. In the proposed protocol, we make sure that BC throughput should maintain a desirable range since each macro mobility event result in a new global transaction. The DTM monitors the BC utilization at the end of every consensus-period. It is done by computing the ratio of total number of new transactions generated in the network by mobile producers to the total number of transaction added in the BC. Note that since all transactions and blocks are broadcast to all GBAs, the utilization computed by all administrators is similar. Two ways can accustom the BC throughput: (i) changing the consensus-period time, it indicates the frequency with which blocks are appended to the BC, and (ii) changing the number of GBAs generating the blocks as each administrator generate a single block within a consensus-period. To better illustrate DTM, let's assume that μ is the BC utilization factor that GBAs calculate at the end of each consensus period. The aim of DTM is to ensure that μ remains in the certain acceptable range to meet the requirements set by network operator (i.e., $\mu_{min} > \mu > \mu_{max}$). From [18], we calculate μ as follow.

[5]The selection of transactions within a block can be random to make trust association more robust.

$$\mu = \frac{N * R_{Tx} * consensus - period}{Tx^{max} * M}, \quad (1)$$

Where N and M denotes the total number of network nodes and the number of nodes assigned with the functionality of BC administrator by the network operator, respectively. R_{Tx} denote the average rate at which BC administrators generate new transactions per second. In particular, R_{Tx} is estimated through a total number of transactions generated within a consensus period. Equation 1 illustrate two ways through which μ can be adjusted to put it in the desired range of μ_{min} and μ_{max}, i.e., changing consensus time period or M. For instance, if μ exceeds the desired maximum range (i.e.,μ_{max}), then DTM first checks that if consensus-period can be reduced. For that DTM compute a new value of consensus-period using Equation 1, which take μ equal to the mid-point of desired range, i.e., between μ_{min} and μ_{max}. It results in a stable operating point in the performance of the network's transaction throughput. Conversely, if the consensus-period cannot be reduced, then the network should be optimized to increase the number of GBAs (i.e., M). The new value for M is also calculated similarly using Equation 1 and by taking the value of μ as a mid-point of the desired range. Besides, the consensus-period is used with the maximum default value while calculating M. This feature allows BlockAuth to scale efficiently for later optimizations. With an increase in the number of M, the transaction throughput would increase, and the max value of consensus-period can be further utilized for throughput enhancement. In case, when the utilization factor drops to the minimum value, i.e., μ_{min}, the similar, but DTM adopts the opposite approach. First, DTM attempts to increase the consensus-period to optimize μ, else it should decrease the number of GBAs, i.e, M.

B. Local IL Transaction Process

Each cluster is comprised of various mobile producers connected to it. The individual cluster is managed by a Local Immutable ledger Administrator (LIA). The local transactions are encrypted with asymmetric encryption, which is done using a lightweight cryptographic hash function. The process uses a PKI infrastructure similar to global BC. Recall that PKI structure is predefined by network owner using As server and SIM device which stores private keys. In each cluster, the LIA centrally manages the local Immutable Ledger (IL) whose structure is similar to BC, and it processes local transactions that are generated and propagated within a cluster. Also, the global transactions that are generated to or from the cluster are managed by the same CH when a producer initially enters into a cluster.

The local IL records all local and global transactions of the P for which the LIA is the only BC administrator. In particular, if P remains in one cluster, CH will maintain both local IL and global BC transactions of the P. As described earlier, each block in the local IL also contains a block header and a policy header. The block header maintains the hash of the previous block to ensure immutability similar to the global BC, as discussed in Section VI-A. The policy header

is in the form of an Access Control List (ACL), which define rules for processing the local and global transactions. Each producer, while roaming in the same cluster uses local IL transaction ID pair, i.e., Tx_i and $previous\_Tx_i$. In particular, the local IL works similar to the global BC, but once being authenticated in the cluster, the producer use and update only local transaction IDs until it moves to another cluster. The first instant of the local transaction ID refers to the genesis output generated by the As. This is performed by the respective CH (i.e., LIA) when a producer initially joins the cluster. Therefore, the structure of the local transactions in local IL is also similar to a global transaction. P uses the local IL to process seamless handover while roaming within the cluster to reduce communication overhead in the core network. The procedure followed by the P in macro and micro mobility scenarios during prefix authentication mechanism is discussed below in detail.

1) Intra cluster handoff: In this type of handoff, BlockA*uth* exploits the location change of P within the same cluster which is controlled only by CH (i.e., also denoted as soft handoff). As mentioned earlier, each producer and CH extract its value of Tx^{ID} when referring to the switch within the same cluster, i.e., Tx_i^{ID}. The mobile P can identify its attachment to a recent cluster by examining the CH ID. This is achieved during the scan process of handoff. In particular, the handoff procedure starts with a neighbor discovery phase called *Scan*, through which mobile host acquires the cluster ID [53], [1]. In this way, after being authenticated in the same cluster, the producer only updates and send the local Tx_i^{ID} chain for being authenticated again in the same cluster. The local IL is administrated and mined by the respective CH. The CH checks the local Tx_i^{ID} to examine if the P have not changed the cluster and find the immutability in the local IL. The transaction is then processed similarly as described previously. After verification, the local blocks are processed and broadcast to the rest of the Bs within the same cluster.

2) Inter cluster handoff: For intercluster mobility event (also named as hard handoff), P update and utilize the global Tx_j ID pairs, which are being used in the previous cluster. The Bs in the new cluster could not find the immutability in the local IL, and therefore, verify the transaction using global Tx_j ID. The transaction is then processed by the respective CH, and later it is stored into global BC.

VII. Performance Evaluation

In this section, first we present the detailed evaluation of the proposed protocol with qualitative security analysis by considering the adversary model elaborated in Section V-B2. We also present an analysis on the robustness of BlockA*uth* against the vulnerabilities based on European Telecommunications Standards Institute (ETSI) [58] risk analysis criteria. Later, we evaluate the performance of BlockA*uth* on ICN routers concerning metrics such as authentication delay, network's throughput and storage cost. The evaluation mainly focuses on the IU authentication mechanism since it is needed at every mobility event, and it is the most demanding step of BlockA*uth*. We also compare BlockA*uth* with the hash-based verification approach that is adopted in most of the prefix attestation proposals [24]. We consider the processing steps to be the same in both approaches (i.e., both issue an Interest Update that will be verified at each router). In the hash-based verification, an IU carries a hash value, while in BlockA*uth*, an IU brings the security content to verify and initiate a BC transaction.

A. Security analysis

We show, in various scenarios, that an adversary is not able to successfully initiate the IU mechanism for the $prefix$(es) which she does not own. We assume that the adversary can be any node functioning in the network, including BC administrators, CHs, and mobile producers. A legitimate P can also act as an adversary (Adv) to re-use the assigned security credentials to perform prefix hijacking or sybil attacks. Moreover, we assume that Adv is capable of sniffing communications, generate false transactions and blocks, discard legitimate transactions, analyze multiple transactions in an attempt to deanonymize a node, and sign fake transactions to legitimize colluding nodes. In the proposed model, we use standard secure asymmetric encryption, digital signatures, and cryptographic hash functions (e.g., SHA256), which cannot be compromised by the Adv.

1) Mitigating Prefix Hijacking Attack: An adversary can pass the initial registration phase if it can insert false transaction in the BC. For this purpose, a Adv need to have a valid signature along with the $prefix$ for the registration of the interest. The computation of signature is with a unique private key of the P, which is registered with $prefix$ owned by it. Since the private key of P is never transmitted over the network and is always stored on the device(s) given by the network owners to the producers, e.g., SIM card. Therefore, Adv is not able to generate a valid registration interest without knowing the private key of the P. The only case, to pass the initial registration process is to replay a valid initial registration interest. In this regard, if replayed interest has received after the As has already received the valid interest, then the malicious interest is discarded by the As, resulting in failed authentication. It is because the hash pointer for current transaction output requires a new transaction output, which is immutable to the previous transaction as elaborated in Section V-E. The only case in which the replayed interest can pass initial authentication is if the valid interest is being received after the replayed one such as due to network congestion or by exploiting signal jammer. At this stage, Adv must generate a valid IU subsequently to update the forwarding state of the edge routers at each mobility event.

To express a valid IU for $prefix$ of P, the adversary must be able to generate a new valid signature including prefix and PoA information and previously used immutable hash pointer towards the previous transaction, i.e., previous Tx^{ID}. Along with $prefix$ name, the complete security content attached (refer to Section V-E) are computed using signature-based encryption and cryptographic hash function (e.g., SHA256). The security properties of PKI-based blockChain methodology makes it impossible for an adversary to generate a valid IU,

TABLE II
BLOCKA*uth* SECURITY ANALYSIS AGAINST VARIOUS THREATS

Threat	Description	Mitigation
Prefix hijacking attacks	Divert traffic of legitimate users and network towards hijacked addresses (prefixes)	Each router in the network authenticates the prefix (IU) to verify the ownership before updating and forwarding the state of the network (see sections V-D and V-E).
Appending attack	Adversary can compromise a BC administrator to generate false blocks and transactions, it leads to corruption of forwarding information in the network	BC administrator can detect a fake block during the verification step (see Section VI-A2), and therefore, it can identify the malicious BCA.
Denial of Service attack (DoS)	Adversary floods router with fake transactions to overwhelm the node such that it cannot devote any resources to process genuine transactions	Unlike [30], in BlockA*uth* the router executes IU authentication mechanism once for a verification process, and it does not entails the invalid IU in its forwarding state (see Section V-E).
Distributed Denial of Service attacks (DDoS)	Adversary attacks on multiple edge router and BCA to flood the network with fake transactions generated from numerous sources	The use of asymmetric keys between BAs make it impossible for an adversary to initiate the DDoS attack. Besides, each transaction and block in the network is verified by the global consistent and immutable image of BC (see Section VI-A).
Replay attacks	A legitimate producer issues the same IU for prefix authentication from two different access points to corrupt the network forwarding information	BC mitigates the replay attack as a single transaction output is immutable to hash pointers. The router selects only the latest valid IU message from the producer.
Packet discarding attack	The BC administrator or CH discard transactions which are being received to and from the cluster members	A cluster member can change the CH or BC administrator it is associated with if it observes that its transactions are not being processed.
False reputation	Any malicious BC administrator tries to increases its reputation	Other BC administrators can detect false increase during transaction verification.

TABLE III
AUTHENTICATION DELAY

Public key signature based						Hash chain based	
RSA 512-bit	RSA 1024-bit	RSA 2048-bit	DSA 512-bit	DSA 1024-bit	DSA 2048-bit	SHA256	MD5
0.16 μs	0.20 μs	0.34 μs	0.90 μs	2.01 μs	6.25 μs	0.0019 μs	0.0090 μs

which is immutable to BC ledger without knowing the private key, cryptographic hash function, previously used immutable transactions pointers, and previous PoA's information. Please note that, although there are possibilities to use the blockchain technology to improve various network security issues such as confidentiality [62] [63], key management [64], and data integrity [41] at different layers of ICN stack. However, we have only considered the security threats at the networking layer. It mainly includes attacks that disturbs the routing and communication process in the whole network and leads to the overall performance degradation. For example, in [65], we have shown that a network attack on inherent features of ICN could lead to QoE degradation for multimedia users. In this paper, the main aim is to use the blockchain technology for providing a simple, fast, and secure authentication framework for mobile produces. Here, the security is provided concerning the producer authentication process, which could be targeted by one of the attacks that we have discussed in Table II.

In Table II, we summarize specific security attacks to which mobile networks and BC are particularly vulnerable and outline how the proposed protocol defends against them. We also analyze the robustness of BlockA*uth* against each of these attacks and the possibility of the attack to happen based on the European Telecommunications Standards Institute (ETSI) [58] risk analysis criteria.

B. Computational Overhead

To evaluate the computational overhead introduced by BlockA*uth* on the routers, the computation of the time required to perform the IU authentication with both hash-based and signature-based approaches is done. Then, based on the analytical model proposed in [66], we compute the impact of BlockA*uth* on the overall router's throughput with an increase in the producer mobility rate. Note that from the term router throughput, we mean the number of regular interest packets processed by a router excluding the Interest Updates. In particular, we compute the delay occurred for the authentication of mobility messages, i.e., IUs, then considering the delay we compute the router's original throughput with increasing producer mobility.

The time required to authenticate the IU is defined as the sum of the time required to retrieve the related transaction from BC (i.e., security context in case of hash-based approaches) and the time required to authenticate the $prefix$. Considering the fact that the BlockA*uth* uses a private BC, the time required to retrieve the relevant security content, i.e., previous transaction using hash IDs, is negligible[6].

To compare the performance of BlockA*uth*, we evaluate BlockA*uth* and hash-chain based protocols [24] by considering the hardware installed on the Bs. We conduct the evaluation

[6]For hash-based approach security context is the recent hash value with corresponding sequence number [24] [66], [24].

on EPYC 7601-AMD processor as reference hardware, and we get both hash-based and signature verification time from [67] as a benchmark. Table III reports the time required to verify message (59 bytes in size) using both, the hash-based and the public-key signature-based schemes. The security context is stored together with the forwarding state in the corresponding table, and it can be retrieved during the regular lookup using transaction ID[7]. Therefore, in BlockA*uth*, the time delay in authenticating the IU is the signature-based verification time, and it is the only dominating factor. Table III reports the time required for the verification using different public-key signature and hash-based cryptographic schemes, respectively. In particular, it reports the results collected in ECRYPT Benchmarking of Asymmetric Systems (eBATS) for public-key signature systems [67]. To compute authentication delay for each cryptographic scheme, we used median values of cycles that are required to verify 59 bytes with a specified processor. We then analyze the impact of the BlockA*uth* authentication delay on edge router throughput. To this end, we compute the router throughput with increasing producer mobility rate.

To compute the router throughput, we utilize the model illustrated in [66], [24]. Let's assume that σ is the ratio of IUs over the total number of normal packets received at a router's ingress interface. Thus, the router throughput can be defined as a ratio of packets/second (λ), which can be calculated as follow.

$$\lambda = \frac{1-\sigma}{\tau_{process} + (\sigma * \tau_{authentication})}. \quad (2)$$

In Equation 2, $\tau_{authentication}$ is the average authentication delay for verifying the prefix, and $\tau_{process}$ is the average processing time that a router takes for an ordinary packet processing. We consider that maximum throughput of edge router, i.e., Bs, to be 0.50 MB/s. Therefore, $\tau_{process}$ is calculated to be 2μs. To compute the impact of BlockA*uth* on router throughput, we apply the values reported in Table III in Equation 2.

Figure 8 depicts that BlockA*uth* exhibit comparable performance to the hash-chain based verification, in case when there is no significant DoS attack. The result shows that the BlockA*uth* can provide approximately 90%–94% of the router's original throughput (i.e., without IUs authentication), when the mobility rate is up to 5%. In precise, the router can maintain almost 94% of the original throughput with faster encryption schemes (e.g., RSA), when 5% of the received traffic includes IUs, i.e., traffic triggered by mobility events. Similarly, for 10%-15% increase in mobility traffic (i.e., IUs), BlockA*uth* is able to maintain 76% to 90% throughput, the percentage value depends on the cryptographic scheme applied (as shown in Figure 8). For instance, while using RSA, BlockA*uth* is able to maintain edge router throughput (i.e., 88%) and it can perform IU authentication on line-rate when mobility rate is up to 20% (refer to Figure 8). Moreover, even with the most optimal mobility scenario where a router receives 30% of IUs in overall traffic, the maximum throughput achieved by a router is 75% (i.e., with RSA 512-bit). In summary, results report that network designer can choose the most appropriate cryptographic scheme depending on the network conditions which relates to the frequency of mobility events.

[7]The previous transaction ID of the producer is the relevant security context

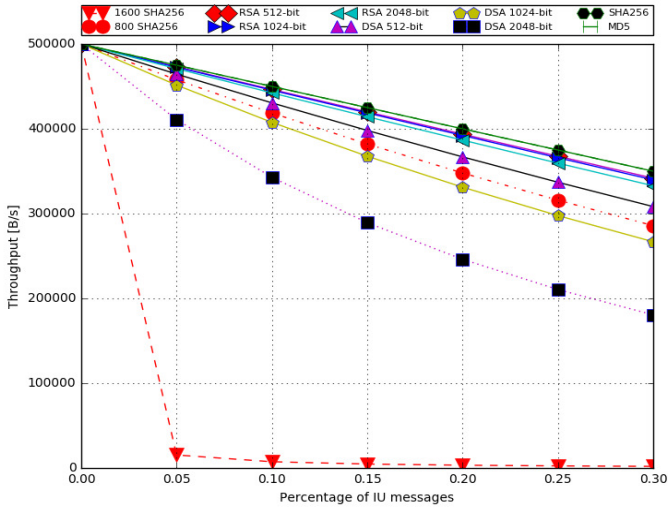

Fig. 8. Edge router throughput

Figure 8 shows the robustness of BlockA*uth* against the DoS attack which is provoked by the one-way hash-chain based prefix authentication proposals [24] (refer to Section IV). We calculated the router throughput for hash-chain based protocol under DoS attack where a router is forced to compute hash (i.e., SHA256) for 800 and 1600 times to detect a legitimate IU and in the process, the throughput decreases significantly as it is shown in Figure 8. The figure depicts noticeable degradation in the throughput where the hash-chain based mechanism (i.e., SHA256) needs to compute just 1600 hashes per IU for authentication. In particular, just with 5% increase in mobility, the router's throughput decreases nearly to zero during a DoS attack. On the other hand, BlockA*uth* efficiently mitigate the DoS attack as it does not require to reach to any synchronized state of the hash chain values.

C. Additional storage cost

The storage cost introduced by the BlockA*uth* relates to the size of the blockchain. Each router is participating in the protocol stores the blockchain along with the forwarding states. After the initial registration of the mobile producer, the size of the BC grows with each new transaction that has been verified and added in it. This relates to the instance when producer issues a legitimate IU at each mobility event. Thus, the additional storage cost introduced by the protocol can be computed as follow.

$$storage\_cost = N_\sigma * size\_Tx. \quad (3)$$

Here N_σ is the number of mobility events initiating IU messages, and $size\_Tx$ is the size of a single transaction needed to perform $prefix$ authentication. For BlockA*uth*, we assume that the size of the Tx is 59 bytes [67] when using both RSA and DSA. For hash chain based authentication, we assume the size of a security context to be 32 bytes [24].

Figure 9 shows an increase in storage cost for the number of mobility events in the network. In general, a mobile EPC network consists of mobile users in the order of 1 million (M). If we consider a scenario where each mobile producer is initiating an IU message, i.e., every producer has triggered a mobility event, then we can observe that the storage cost is about 60 MB for each router. Current routers can easily store such amount of data due to the availability of storage memory. In addition, the network owner can optimize BlockA*uth* by using an efficient data pruning technique.

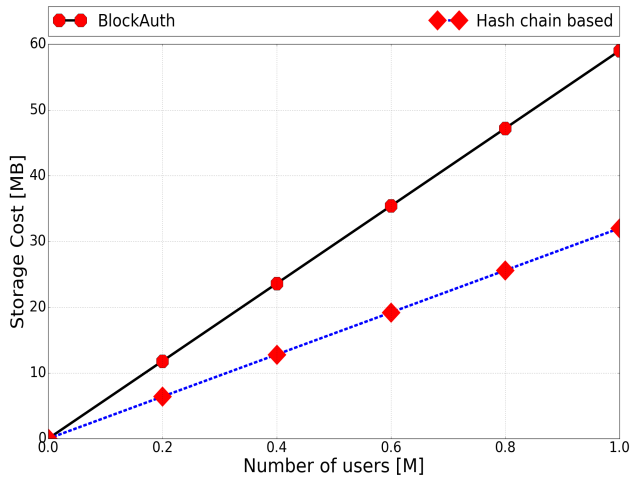

Fig. 9. Additional storage cost at each router

VIII. Conclusion and Future Directions

In the context of ICN, the practice of producer/consumer communication model primarily appreciates seamless mobility support to mobile nodes. It is due to the result of decoupling time and space among request resolution and content transfer. However, the dynamic interaction between producer and forwarding plane in ICN introduces new security challenges in the network. For instance, a producer could send a false interest update request in the network, and all the ICN routers that receive such a request will update their forwarding table accordingly, thus leave the forwarding table in an inconsistent state. In this paper, we investigated and proposed solutions to the security challenges related to producer mobility tracing-based protocols. Particularly, to mitigate prefix hijacking attacks and resolve security and privacy issues in ICN mobility management, we presented an efficient blockchain based distributed prefix authentication protocol, which offers reliable and faster mobile user authentication. We show that the proposed protocol is completely distributed, lightweight, and it can be easily deployed on different network access platforms (e.g., 4G, 5G, and WiFi). The security and performance analysis shows that the proposed protocol performs significantly better when compared to state-of-the-art, and it efficiently mitigates prefix hijacking, Denial of Service, and other telecommunication networking related attacks. Besides, the proposed approach can maintain the router's original throughput up to 94% (i.e., able to perform prefix authentication at line rate). In terms of storage, it can handle billion of mobile producers just by consuming tens of megabyte on each router.

In this paper, we have proposed a blockchain-based ICN framework and showed its deployment feasibility and working efficiency by evaluating it against metrics such as delay in producer authentication, routers throughput, and storage cost. We have also showed that the proposed framework could handle various networking attacks (please refer to Table II). However, a detailed analysis showing the level of resistance that BlockAuth provides against these attacks still lacks and will be taken into account as a future work. Further, BlockAuths privacy is yet to be evaluated. For example, the blockchain stores the mobility history of all producers, thus, once an attacker has access to the blockchain data it could exploit the information stored in it to breach producers privacy. In future, we will also try to find out that how the BlockAuth architecture behaves in presence of blockchain forks and how we can reduce the rate of these forks in the network. Analyzing the impact of forks is important as it could lead to substantial divergences in the authentication mechanism that are in-place to perform fast producer authentication. Finally, we believe that blockchain could provide a set of new opportunities that could improve the overall performance of ICN. It can be done by using blockchain technology for improving various areas such as routing, namespacing, and key management.

Acknowledgement

Mauro Conti is supported by a Marie Curie Fellowship funded by the European Commission (agreement PCIG11-GA-2012-321980). This work is also partially supported by the EU LOCARD Project (agreement H2020-SU-SEC-2018-832735). This work is partially supported by the grant n. 2017-166478 (3696) from Cisco University Research Program Fund and Silicon Valley Community Foundation.

References

[1] C. E. Perkins, "Mobile IP," *IEEE Communications Magazine*, vol. 35, no. 5, pp. 84–99, 1997.

[2] A. G. Valkó, "Cellular IP: a new approach to Internet host mobility," *ACM SIGCOMM CCR*, vol. 29, no. 1, pp. 50–65, 1999.

[3] S. Das *et al.*, "TeleMIP: telecommunications-enhanced mobile IP architecture for fast intradomain mobility," *IEEE Personal Communications*, vol. 7, no. 4, pp. 50–58, 2000.

[4] C. Anastasiades, T. Braun, and V. A. Siris, *Information-Centric Networking in Mobile and Opportunistic Networks*. Cham: Springer International Publishing, 2014, pp. 14–30. [Online]. Available: https://doi.org/10.1007/978-3-319-10834-6_2

[5] C. Fang, H. Yao, Z. Wang, W. Wu, X. Jin, and F. R. Yu, "A Survey of Mobile Information-Centric Networking: Research Issues and Challenges," *IEEE Communications Surveys Tutorials*, pp. 1–1, 2018.

[6] Y. Zhang, A. Afanasyev, J. Burke, and L. Zhang, "A survey of mobility support in Named Data Networking," in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2016, pp. 83–88.

[7] J. Augé *et al.*, "Anchor-less Producer Mobility in ICN," in *ACM Proceedings of the 2nd International Conference on Information-Centric Networking*, 2015, pp. 189–190.

[8] D. Han, M. Lee, K. Cho, T. Kwon, and Y. Choi, "Publisher mobility support in content centric networks," in *The International Conference on Information Networking 2014 (ICOIN2014)*, Feb 2014, pp. 214–219.

[9] D.-h. Kim, J.-h. Kim, Y.-s. Kim, H.-s. Yoon, and I. Yeom, "Mobility Support in Content Centric Networks," in *Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking*, ser. ICN '12. New York, NY, USA: ACM, 2012, pp. 13–18. [Online]. Available: http://doi.acm.org/10.1145/2342488.2342492

[10] L. Wang, O. Waltari, and J. Kangasharju, "MobiCCN: Mobility support with greedy routing in Content-Centric Networks," in *2013 IEEE Global Communications Conference (GLOBECOM)*, Dec 2013, pp. 2069–2075.

[11] Y. Zhang, H. Zhang, and L. Zhang, "Kite: A mobility support scheme for ndn," in *ACM Conference on Information-Centric Networking*. ACM, 2014, pp. 179–180.

[12] H. Ballani, P. Francis, and X. Zhang, "A Study of Prefix Hijacking and Interception in the Internet," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 265–276, Aug. 2007. [Online]. Available: http://doi.acm.org/10.1145/1282427.1282411

[13] M. Al-Shurman, S.-M. Yoo, and S. Park, "Black Hole Attack in Mobile Ad Hoc Networks," in *Proceedings of the 42Nd Annual Southeast Regional Conference*, ser. ACM-SE 42. New York, NY, USA: ACM, 2004, pp. 96–97. [Online]. Available: http://doi.acm.org/10.1145/986537.986560

[14] C. Ghali, "Needle in a Haystack : Mitigating Content Poisoning in Named-Data Networking," 2014.

[15] M. Conti, P. Gasti, and M. Teoli, "A Lightweight Mechanism for Detection of Cache Pollution Attacks in Named Data Networking," vol. 57, no. 16. New York, NY, USA: Elsevier North-Holland, Inc., Nov. 2013, pp. 3178–3191. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2013.07.034

[16] E. C. Ferrer, "The blockchain: a new framework for robotic swarm systems," *CoRR*, vol. abs/1608.00695, 2016, last access July 2019. [Online]. Available: http://arxiv.org/abs/1608.00695

[17] A. Dorri, S. S. Kanhere, and R. Jurdak, "Towards an Optimized BlockChain for IoT," in *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*, April 2017, pp. 173–178.

[18] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "LSB: A Lightweight Scalable BlockChain for IoT Security and Privacy," *CoRR*, vol. abs/1712.02969, 2017, last access July 2019. [Online]. Available: http://arxiv.org/abs/1712.02969

[19] S. H. Hashemi, F. Faghri, and R. H. Campbell, "Decentralized User-Centric Access Control using PubSub over Blockchain," *CoRR*, vol. abs/1710.00110, 2017, last access July 2019. [Online]. Available: http://arxiv.org/abs/1710.00110

[20] A. Moinet, B. Darties, and J. Baril, "Blockchain based trust & authentication for decentralized sensor networks," *CoRR*, vol. abs/1706.01730, 2017, last access July 2019. [Online]. Available: http://arxiv.org/abs/1706.01730

[21] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, "BlockChain: A Distributed Solution to Automotive Security and Privacy," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 119–125, DECEMBER 2017.

[22] A. de la Rocha Gómez-Arevalillo and P. Papadimitratos, "Blockchain-based Public Key Infrastructure for Inter-Domain Secure Routing," in *International Workshop on Open Problems in Network Security (iNetSec)*, ser. Open Problems in Network Security, J. Camenisch and D. Kesdoğan, Eds., vol. IFIP eCollection-1, Rome, Italy, May 2017, pp. 20–38. [Online]. Available: https://hal.inria.fr/hal-01684192

[23] S. Raju, S. Boddepalli, S. Gampa, Q. Yan, and J. S. Deogun, "Identity management using blockchain for cognitive cellular networks," in *2017 IEEE International Conference on Communications (ICC)*, May 2017, pp. 1–6.

[24] A. Compagno, X. Zeng, L. Muscariello, G. Carofiglio, and J. Augé, "Secure Producer Mobility in Information-Centric Network," in *Proceedings of the 4th ACM Conference on Information-Centric Networking*, ser. ICN '17. New York, NY, USA: ACM, 2017, pp. 163–169. [Online]. Available: http://doi.acm.org/10.1145/3125719.3125725

[25] G. Xylomenos, C. N. Ververidis, V. A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. V. Katsaros, and G. C. Polyzos, "a survey of information-centric networking research," *IEEE Communications Surveys Tutorials*.

[26] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, and B. Ohlman, "A survey of information-Centric networking," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 26–36, July 2012.

[27] G. Xylomenos, X. Vasilakos, C. Tsilopoulos, V. A. Siris, and G. C. Polyzos, "Caching and mobility support in a publish-subscribe internet architecture," *IEEE Communications Magazine*, vol. 50, no. 7, pp. 52–58, July 2012.

[28] L. Zhang *et al.*, "Named Data Networking," *ACM SIGCOMM CCR*, vol. 44, no. 3, pp. 66–73, 2014.

[29] V. Jacobson *et al.*, "Networking Named Content," in *ACM International Conference on Emerging Networking Experiments and Technologies*, 2009, pp. 1–12.

[30] A. Compagno, M. Conti, and M. Hassan, *An ICN-Based Authentication Protocol for a Simplified LTE Architecture*. Cham: Springer International Publishing, 2018.

[31] Z. Zhang, C.-H. Lung, I. Lambadaris, and M. St-Hilaire, "When 5G meets ICN: An ICN-based caching approach for mobile video in 5G networks," *Computer Communications*, vol. 118, pp. 81 – 92, 2018.

[32] R. Ravindran, A. Chakraborti, S. O. Amin, A. Azgin, and G. Wang, "5G-ICN: Delivering ICN Services over 5G Using Network Slicing," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 101–107, May 2017.

[33] Y. Nishiyama, M. Ishino, Y. Koizumi, T. Hasegawa, K. Sugiyama, and A. Tagami, "Proposal on routing-based mobility architecture for ICN-based cellular networks," in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2016, pp. 467–472.

[34] R. Ravindran, P. Suthar, A. Chakraborti, S. O. Amin, A. Azgin, and G. Wang, "Deploying ICN in 3GPPs 5G NextGen Core Architecture," in *2018 IEEE 5G World Forum (5GWF)*, July 2018, pp. 26–32.

[35] M. Amadeo, C. Campolo, A. Iera, and A. Molinaro, "Named data networking for IoT: An architectural perspective," in *2014 European Conference on Networks and Communications (EuCNC)*, June 2014, pp. 1–5.

[36] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts," in *2016 IEEE Symposium on Security and Privacy (SP)*, May 2016, pp. 839–858.

[37] D. G. Wood, "Ethereum: a Secure Decentralised Generalised Transaction Ledger," 2016.

[38] G. Brambilla, M. Amoretti, and F. Zanichelli, "Using Block Chain for Peer-to-Peer Proof-of-Location," *CoRR*, vol. abs/1607.00174, 2016, last access July 2019. [Online]. Available: http://arxiv.org/abs/1607.00174

[39] M. Vukolić, "The Quest for Scalable Blockchain Fabric: Proof-of-Work vs. BFT Replication," in *Open Problems in Network Security*, J. Camenisch and D. Kesdoğan, Eds. Cham: Springer International Publishing, 2016, pp. 112–125.

[40] G. Zyskind, O. Nathan, and A. Pentland, "Decentralizing Privacy: Using Blockchain to Protect Personal Data," in *2015 IEEE Security and Privacy Workshops (SPW)*, vol. 00, May 2015, pp. 180–184. [Online]. Available: doi.ieeecomputersociety.org/10.1109/SPW.2015.27

[41] N. Fotiou and G. C. Polyzos, "Decentralized name-based security for content distribution using blockchains," in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2016, pp. 415–420.

[42] A. T. Campbell, J. Gomez, S. Kim, A. G. Valko, C.-Y. Wan, and Z. R. Turanyi, "Design, implementation, and evaluation of cellular IP," *IEEE Personal Communications*, vol. 7, no. 4, pp. 42–49, Aug 2000.

[43] S. Das, A. Misra, and P. Agrawal, "TeleMIP: telecommunications-enhanced mobile IP architecture for fast intradomain mobility," *IEEE Personal Communications*, vol. 7, no. 4, pp. 50–58, Aug 2000.

[44] "A survey of BGP security issues and solutions," *Proceedings of the IEEE*, vol. 98, no. 1, pp. 100–122, 1 2010.

[45] G. Huston, M. Rossi, and G. Armitage, "Securing BGP x2014; A Literature Survey," *IEEE Communications Surveys Tutorials*, vol. 13, no. 2, pp. 199–222, Second 2011.

[46] S. L. Murphy, M. R. Badger, and B. Wellington, "OSPF with Digital Signatures," *RFC*, vol. 2154, pp. 1–29, 1997.

[47] T. Wan, E. Kranakis, and P. C. van Oorschot, "S-RIP: A Secure Distance Vector Routing Protocol," in *Applied Cryptography and Network Security*, M. Jakobsson, M. Yung, and J. Zhou, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 103–119.

[48] S. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol (S-BGP)," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 4, pp. 582–592, April 2000.

[49] R. White, "Securing BGP through secure origin BGP," vol. 6, 01 2003.

[50] R. Langar, N. Bouabdallah, and R. Boutaba, "Mobility-aware Clustering Algorithms with Interference Constraints in Wireless Mesh Networks," *Comput. Netw.*, vol. 53, no. 1, pp. 25–44, Jan. 2009. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2008.09.012

[51] C. R. Lin and M. Gerla, "Adaptive clustering for mobile wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1265–1275, Sep 1997.

[52] M. Hajjar, G. Aldabbagh, and N. Dimitriou, "Using clustering techniques to improve capacity of LTE networks," in *2015 21st Asia-Pacific Conference on Communications (APCC)*, Oct 2015, pp. 68–73.

[53] J. S. Thainesh, N. Wang, and R. Tafazolli, "Reduction of core network signalling overhead in cluster based LTE small cell networks," in *2015 IEEE 20th International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD)*, Sept 2015, pp. 226–230.

[54] M. Chatterjee, S. K. Das, and D. Turgut, "An on-demand weighted clustering algorithm (WCA) for ad hoc networks," in *Global Telecommunications Conference, 2000. GLOBECOM '00. IEEE*, vol. 3, Nov 2000, pp. 1697–1701 vol.3.

[55] Y. Zhang, H. Zhang, and L. Zhang, "Kite: A Mobility Support Scheme for NDN," in *Proceedings of the 1st ACM Conference on Information-Centric Networking*, ser. ACM-ICN '14. New York, NY, USA: ACM, 2014, pp. 179–180. [Online]. Available: http://doi.acm.org/10.1145/2660129.2660159

[56] K. D. Dimou, M. Wang, Y. Yang, M. Kazmi, A. Larmo, J. Pettersson, W. Mller, and Y. Timner, "Handover within 3GPP LTE: Design Principles and Performance." in *VTC Fall*. IEEE, 2009, pp. 0–. [Online]. Available: http://dblp.uni-trier.de/db/conf/vtc/vtc2009f.html#DimouWYKLPMT09

[57] 3rd Generation Partnership Project, "3GPP System Architecture Evolution (SAE)," in *"Technical Specification Group Services and System Aspects"*, 2008.

[58] ETSI TS 102 165-1 V4.2.3 (2011-03), "Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN); Methods and protocols; Part 1: Method and proforma for Threat, Risk, Vulnerability Analysis," 2011.

[59] C. Dukkipati, Y. Zhang, and L. C. Cheng, "Decentralized, BlockChain Based Access Control Framework for the Heterogeneous Internet of Things," in *Proceedings of the Third ACM Workshop on Attribute-Based Access Control*, ser. ABAC'18. New York, NY, USA: ACM, 2018, pp. 61–69. [Online]. Available: http://doi.acm.org/10.1145/3180457.3180458

[60] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system, http://bitcoin.org/bitcoin.pdf," 2008.

[61] C. Fromknecht, D. Velicanu, and S. Yakoubov, "A Decentralized Public Key Infrastructure with Identity Retention," *IACR Cryptology ePrint Archive*, vol. 2014, p. 803, 2014.

[62] K. Zhu, Z. Chen, W. Yan, and L. Zhang, "Security attacks in named data networking of things and a blockchain solution," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4733–4741, June 2019.

[63] H. Li, K. Wang, T. Miyazaki, C. Xu, S. Guo, and Y. Sun, "Trust-enhanced content delivery in blockchain-based information-centric networking," *IEEE Network*, pp. 1–7, 2019.

[64] M. Labbi, N. Kannouf, Y. Chahid, M. Benabdellah, and A. Azizi, "Blockchain-based pki for content-centric networking," in *Innovations in Smart Cities Applications Edition 2*, M. Ben Ahmed, A. A. Boudhir, and A. Younes, Eds. Cham: Springer International Publishing, 2019, pp. 656–667.

[65] M. Conti, R. Droms, M. Hassan, and C. Lal, "Fair-RTT-DAS: A Robust and Efficient Dynamic Adaptive Streaming over ICN," *Computer Communications*, vol. 129, pp. 209 – 225, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0140366418303372

[66] C. Ghali, M. A. Schlosberg, G. Tsudik, and C. A. Wood, "Interest-Based Access Control for Content Centric Networks," in *Proceedings of the 2Nd ACM Conference on Information-Centric Networking*, ser. ACM-ICN '15. New York, NY, USA: ACM, 2015, pp. 147–156. [Online]. Available: http://doi.acm.org/10.1145/2810156.2810174

[67] D. J. Bernstein and T. Lange, "eBACS: ECRYPT Benchmarking of Cryptographic Systems," https://bench.cr.yp.to/.