Towards Secure Network Coding Enabled Mobile Small Cells

Vipindev Adat, Tafseer Akhtar, Ilias Politis, Christos Tselios, Stavros Kotsopoulos
Wireless Communications Laboratory
University of Patras, Greece
{vipindev, tafseer, ipolitis, tselios, kotsop}@ece.upatras.gr

Abstract—As the validation and trials of fifth-generation communication architectures are maturing, it becomes evident that efficient bandwidth utilization, security, and seamless cooperation among heterogeneous small cell access networks are among the vital requirements that the future wireless networks must satisfy. A potential candidate technology that can ensure bandwidth efficiency and high resilient communications, especially in the adverse wireless networking conditions is network coding. However, without a security strategy for protecting the data and the identity of the communication entities, network coding can be proved as a factor of severely degrading networking performance. Although several cryptography-based schemes have been used over the recent years to ensure secure communications over network coding enabled networks, none was specifically designed to address the mobile small cell access network of the forthcoming fifth-generation paradigm. This paper proposes a secure network coding architecture for mobile small cells, which due to their unpredictable topology, render the most encryption key management schemes unsuitable and costly in terms of signaling. The proposed secure network coding architecture incorporates blockchains in an effort to ensure efficient key distribution and reduced signaling overhead.

I. INTRODUCTION

Future communication networks are expected to be a heterogeneous environment with hard constraints on throughput and energy efficiency. The rapid evolution in devices with digital identity has created a heterogeneous network of smart mobile devices with very diverse requirements in terms of quality of service and security [1]. Explosive growth in the number of connected devices and data volume, high data rate, and very low latency are few of the essential features of 5G network environment, which will result in movement of the network from base station-centric to device-centric. This movement is a major paradigm shift, which demands the deployment of small cells in a highly dense manner with user-centric design, as the future user will store, relay, compute and transmit data traffic as well. Highly dense deployment of various small cells creates a heterogeneous network which also includes macro cell and relay nodes. Hyperdense heterogeneous networks (HetNets) is an integral part 5G environment, and it also brings various inherent challenges which will outclass the current network capabilities in terms of overall end-user quality of experience [2]. Recent studies and early evaluation pilots [3], [4] on the emerging 5G paradigm have indicated some of the key technologies [5] which will form an integral part of the next-generation wireless networks. Sneak peeks to the 5G networks are pointing towards the high data rate, low latency communications with minimal energy spent. It is expected that 5G will address a dense heterogeneous small cell environment along with diverse radio access technologies (like 3G, 4G, 5G, etc.), partially focusing on Multi-access Edge Computing (MEC) and Fog Computing deployment architectures [6]. The 5G architecture addressing the heterogeneous mobile devices using the small cell structure and device to device communication is also adaptable to new technologies to improve the quality of experience. Network coding, a method to optimize the throughput of a network, is being considered as one of them to improve the throughput and resilience of the future networks [7].

Network coding (NC) is a breakthrough technology that can be used for increasing the resiliency and throughput of a network. The communications in a wireless environment are mostly broadcast in nature where multiple replicas of the same transmission can be received at multiple destinations, although every distinct pair of nodes has spatial diversity through independent channel fading effect. The advantage of the broadcast nature of wireless environment can be taken by capturing the multiple copies of transmission signals from relays sent by sources. These copies of same transmission signals can be repeatedly sent from relays, apart from receiving the transmission signals from the source through a probable direct link, the destination can receive the same signals from various linked relays, which will significantly improve the overall communication performance as shown in Fig. 1. These spatial diversities can be utilized through cooperative communication (CC) to handle the wireless channel fading which affects the performance of overall wireless communication. Traditional cooperation method uses the simple process and forward approach, whereas, network coded cooperating (NCC) approach [8] uses NC that allows encoding of collected data from various sources at relays before forwarding, and these network coded data are then decoded at the corresponding destination. The random linear network coding (RLNC) was proposed as an effective method to improve the throughput and efficient utilization of the bandwidth available for wireless networks [9]. It ensures the harness of available bandwidth and provides high resiliency. But the network coding is also

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 722424
susceptible to security threats like pollution attacks which can have drastic effects on the efficiency of the whole system [10]. Since the packets will be mixed at the intermediate nodes, a single polluted packet can pollute all the packets which it will be mixed with during its transition and this makes pollution attack in a network coded environment as a devastating attack. This issue has been addressed with various solutions in the literature. Dedicated end-to-end frameworks do exist [11], but cryptographic solutions based on message authentication codes [12] against pollution attacks are considered as a low complexity solution with lower latency and computational requirements. However, it requires key sharing schemes to ensure the proper verification of the MACs. The network coded small cells will require these keys to be shared between different small cells in order to perform a secure network coding between the small cells. In a dense mobile small cell environment it is not easy to have pre-shared keys between all of them to facilitate the communication. In this paper, the aim is to address this problem and propose a solution using blockchains and in particular BigchainDB, which satisfies the requirements for low latency and minimum processing capacity.

The rest of the paper is organized as follows. Section II shows the catastrophic effect of pollution attacks in RLNC networks and shows the specific case of network coding enabled mobile small cells. A general MAC scheme and how the blockchain-based key sharing scheme will extend the original scheme to the small cell environment are shown in section III. Analysis of the scheme is presented in section IV and followed by concluding remarks.

II. RLNC AND POLLUTION ATTACKS

RLNC based networks are susceptible to pollution attacks in which malicious users try to inject corrupted packets to the network and it will have a catastrophic effect on the communication. Since the packets are mixed on the fly, a single corrupted packet can possibly corrupt many packets if it proceeds further down the path. The cryptographic techniques used in traditional networks would not be sufficient for preventing pollution attack since the genuine packets will also get modified during the transition. Integrity schemes based on Homomorphic signatures and message authentication codes (MACs) are proposed to address this issue. Integrity schemes based on Homomorphic MACs have been widely used against pollution attacks with low computational and communication overhead over the system. However, the MAC scheme requires some pre-distributed keys at each node which enable them to verify the incoming packets against their tags. In traditional examples of RLNC based networks and the state of the art integrity schemes, these keys are supposed to be pre-distributed offline or during the setup phase. In the future networks involving small cells, it may not be the case to have shared keys beforehand between different users associated with different small cells. Further, the mobile nodes and mobile small cells, which is a highly possible and unavoidable scenario in the 5G and beyond 5G environment, having shared keys in advance may not be possible. In the conventional small cell architectures, the elected lead or the access point in the small cell will be in charge of authenticating the keys (or renewing them) of all the nodes in the small cells by itself or in coordination with the security authority. Cooperation between these small cells is essential to facilitate secure network coding enabled environment. The straightforward way of this key sharing is by using the public-private keys of the nodes in communication to create a secure channel and share the keys. However, this creates a huge overhead since to make it practically possible, each small cell will require a secure channel with each other small cell in the network, as shown in Fig. 2. Also, it delays the whole process. This paper addresses this problem with the help of blockchain technology.

III. MAC BASED INTEGRITY SCHEME FOR NETWORK CODED SMALL CELLS

A. Homomorphic MAC for Secure Network Coding

MAC-based authentication schemes are widely accepted as one of the efficient approaches to prevent pollution attacks in network coded environments. This cryptographic approach uses the symmetric key distribution for creating and verifying the tags (MACs). There have been quite a lot of approaches based on MACs for ensuring integrity. Most of the approaches follow the same structure of algorithms and computations and differ in the key sharing schemes or other aspects to improve efficiency. Homomac [12] was one of the pure and initial works using MAC-based integrity scheme for secure network coding. However, it was susceptible to tag pollution attacks.
between nodes in different small cells and spread over a large geographical area. In the initial set up phase, keys will be distributed by the security agency to the nodes via the mobile small cell hotspot (or access point or cluster head) in each small cell. However, the MAC-based integrity schemes require at least a subset of the keys owned by the source to be known by each forward and receiving node in the network. In typical cases, the only way to ensure this is by creating a secure channel between all the participating nodes and shares the key using this channel. However, this will create a large overhead since the source node will have to create separate channels for each participating nodes and share the keys. It will create not only computational overhead over the source node but also a delay in the initial setup phase. If there are multiple nodes trying to communicate, the situation will get unacceptably worse. This problem can be addressed by a blockchain-based database shared between the small cells. Each source node while initiating a communication, will also create a transaction to be added to the block as shown in the format shown in Table I. The transaction format includes the address of final destination node(s) and the keys used by the source to create the tags along with its identity ensuring fields such as transaction hash and an identification number. The transaction hash will be created by the source node using its private key and will enable the other nodes to verify that the transaction is submitted by the original source node itself.

In the practical scenario, it is possible that more than one node will be transmitting during the same time and in such cases more than one transaction will be created at approximately the same time. These transactions occurring at a small time period are collected into a candidate block and it will distributed to the network of small cell access points for validation. The candidate block will be verified by the proof of stake method by the participants in the block validation. In general, blocks are mined by a proof of work algorithm which involves the computation of a nonce satisfying hard criteria and the participant nodes will compete for creating the proper nonce for a block. However, this is highly energy inefficient and resource exploiting method since computing this hard nonce itself is a costly task and the competition results in wasting all the energy and resources used by the nodes which were not successful in verifying the block for the first time. Energy efficiency is an important requirement for the future network design and thus we cannot afford such an under-utilization of energy and resources. Considering this constraint, we propose a more efficient proof of stake algorithm to verify the blocks in the chain.

Thus Ripple [13] protocol was proposed which protects the network from tag pollution attack as well with the time asymmetry ensured by the Ripple transmission protocol. Further MacSig [14] was a pure cryptographic approach, which uses the symmetric key based MACs and asymmetric key-based signatures to ensure the integrity of data and tags respectively. Esfahani et al. improved the previous proposals in [15], [16] to be more efficient in terms of computational and communication overhead by employing c-cover free based key distribution. In [17], [18], authors were looking at the different approaches on sharing the tags in a network coded small cell environment in order to reduce the bandwidth overhead over the communication channels. All these approaches address the message authentication using the same kind of cryptographic techniques and assume some pre-shared secret keys available at the nodes (other than Ripple where the keys are shared with some time asymmetry). However, none of the previous approaches address the problem of small cell environment mentioned in Section II. We propose the key sharing scheme using blockchains for facilitating the secure network coded mobile small cell environment.

B. Blockchain based key sharing scheme for network coding enabled small cells

Blockchain is a distributed ledger shared between all the participating entities and records the blocks in a chronological order ensured by cryptography [19]. Blockchain technology was there from a longer time, but it came into the highlight with the bitcoin as the core behind the crypto currency [20]. It enables a group of users to reach a consensus regarding the transactions happened in the time period from previous block to the new block generation and it’s applications other than the cryptocurrency is being explored widely [21] in the recent days. Blockchain itself is considered as a secure distributed ledger since each block is created with a hash to the previous block and thus modifying an existing block is a very high cost computational operation. Blockchain technology is primarily used in crypto currencies even though a secure distributed ledger could have substantial improvements in various applications like smart contracts and intelligent systems [22]. In this work, we are trying to explore the suitability of a blockchain based key sharing scheme for secure mobile small cells.

In the network coding enabled mobile small cell network depicted in Fig. 3 all small cells form a blockchain over the network. This blockchain act as a distributed and decentralized database of the small cells. The proposed architecture is based on bigchainDB [23], a decentralized database having blockchain properties. Each bigchainDB node wraps up a rethinkDB node, which is an existing distributed database system [24], as shown in the figure. So in our proposed model, every small cell head will have a rethinkDB node which communicates with the other similar nodes in order to create a distributed database. A MAC based integrity scheme is deployed over the D2D communication links inside the small cells and the communication may also take place between nodes in different small cells and spread over a large geographical area. In the initial set up phase, keys will be distributed by the security agency to the nodes via the mobile small cell hotspot (or access point or cluster head) in each small cell. However, the MAC-based integrity schemes require at least a subset of the keys owned by the source to be known by each forward and receiving node in the network. In typical cases, the only way to ensure this is by creating a secure channel between all the participating nodes and shares the key using this channel. However, this will create a large overhead since the source node will have to create separate channels for each participating nodes and share the keys. It will create not only computational overhead over the source node but also a delay in the initial setup phase. If there are multiple nodes trying to communicate, the situation will get unacceptably worse. This problem can be addressed by a blockchain-based database shared between the small cells. Each source node while initiating a communication, will also create a transaction to be added to the block as shown in the format shown in Table I. The transaction format includes the address of final destination node(s) and the keys used by the source to create the tags along with its identity ensuring fields such as transaction hash and an identification number. The transaction hash will be created by the source node using its private key and will enable the other nodes to verify that the transaction is submitted by the original source node itself.

In the practical scenario, it is possible that more than one node will be transmitting during the same time and in such cases more than one transaction will be created at approximately the same time. These transactions occurring at a small time period are collected into a candidate block and it will distributed to the network of small cell access points for validation. The candidate block will be verified by the proof of stake method by the participants in the block validation. In general, blocks are mined by a proof of work algorithm which involves the computation of a nonce satisfying hard criteria and the participant nodes will compete for creating the proper nonce for a block. However, this is highly energy inefficient and resource exploiting method since computing this hard nonce itself is a costly task and the competition results in wasting all the energy and resources used by the nodes which were not successful in verifying the block for the first time. Energy efficiency is an important requirement for the future network design and thus we cannot afford such an under-utilization of energy and resources. Considering this constraint, we propose a more efficient proof of stake algorithm to verify the blocks in the chain.

| Field                | Description                                      |
|----------------------|--------------------------------------------------|
| Transaction Number   | Identification number of the transaction         |
| Transaction Hash     | A hash of the transaction                        |
| Source node address  | Address of the source node                       |
| Destination Small Cell| Address of the destination small cell            |
| Keys                 | Keys used by source nodes                       |

| TRANSACTION FORMAT   | TABLE I                                         |
|----------------------|-------------------------------------------------|
| Field                | Description                                      |
|----------------------|--------------------------------------------------|
| Transaction Number   | Identification number of the transaction         |
| Transaction Hash     | A hash of the transaction                        |
| Source node address  | Address of the source node                       |
| Destination Small Cell| Address of the destination small cell            |
| Keys                 | Keys used by source nodes                       |
In proof of stake algorithm [25], there will be no competition for verifying the block, but all the participating nodes will agree on one node to do the verification based on the stakes it holds. The stake can be anything based on which a fare selection can be made. It should be taken care that no single or group of participants will be able to completely take the authority for performing the verification process. In our case, we propose to use the entries in the destination address of transactions to decide the verifier (commonly called as the forger in proof of stake based blockchains). The small cell which is supposed to be served as the destination for maximum number of transactions can forge the candidate block and verify it to be the part of the blockchain. The verification process are always assigned to the small cell head in order to ensure there will be no lack of resource power to do it. Further, the POS process doesn’t look for a very hard computational nonce since here the security is not based on the computational complexity of the nonce. If the forger did perform maliciously, it should be removed from the network to ensure the security of the system. Since the forger is supposed to receive maximum of the messages, it should ensure the keys (transactions) are properly shared between all the participating nodes. This fortify the security of the proof of stake based blockchain. In BigchainDB, this consensus is achieved by Tendermints Byzantine Fault Tolerance (BFT) algorithm, which is a PoS based concept. Each block will have a number of transactions which in turn will have the keys used by the source nodes to create tags during that time. Each block will also have the time-stamp and since it will be shared with all the small cells, now any participating node can access the required keys from the block. The block format is shown in Table II.

### Table II

| Field             | Description                                      |
|-------------------|--------------------------------------------------|
| Version Number    | Block Version Number                             |
| Previous block hash | A hash of the previous block                     |
| Merkle Tree Root  | Hash of the Merkle tree root of transactions     |
| Timestamp         | Timestamp of the block                           |
| Nonce             | A counter for the Proof of stake                 |
| Forger Address    | Address of the small cell who verified the block |
| **Block Payload** |                                                  |
| Transaction 1     | The first transaction in this block              |
| ...               |                                                  |
| Transaction N     | The Nth transaction in this block                |

Table II.

### IV. PROPOSED BLOCKCHAIN-BASED SECURED NC

Even though many key management and distribution schemes are proposed for network coded wireless networks, the specific problem of key management for network coded mobile small cells is not addressed in the literature to the best of our knowledge. The conventional method of sharing the keys using secure channel may not be the suitable option due to the large overhead it can create. Each source node will have to create a secure channel with all the nodes which will be in the flow of the packet towards the destination. In the worst case, every small cell will require secure channels with all other small cells for key sharing as in Fig. 2. In a mobile environment, such a condition is practically impossible or extremely costly in terms of bandwidth requirements and signalling. Furthermore, the delay in calculating the path and
creating secure connection between each intermediate and destination node will force to a large initial delay. It may also create a large signalling cost depending on the number of intermediate nodes. Fig. 3 illustrates a network of small cells which works to ensure a secure blockchain based key sharing which will tackle both the problems.

In this blockchain based key sharing scheme, most of these challenges are addressed in an affordable way in terms of computational and signalling cost as well as ensuring low latency in the initialization process. Since the keys are verified by a proof of stake algorithm involving the small cells and then added as a block in the chain available to all the small cells, it does not require to calculate the path beforehand which eliminate the initial latency in the system and suites more to a mobile environment. The computational cost of the scheme can also be more efficient than the conventional scheme depending on the number of participants or intermediate nodes requiring the keys. In our scheme, the computational cost of verifying a key is independent of the number of beneficiaries but depend on the signing by the source and the verification procedure of the blockchain. Signing by the source is necessary even in the conventional scheme at least once and as many as the number of beneficiaries. The verification cost depends on the hardness of calculating the nonce and hash of the previous block. Since the bigchainDB uses a BFT consensus algorithm, it is not necessary to make the calculation computationally as costly as a PoW based blockchain to ensure security. It is also to be noted that once the verification is done, any small cell in the network will be able to securely access the keys so the efficiency of the scheme increases as the number of inter cell intermediate nodes increases. Further, even if there are more than one small cells initiating the communication during a block time, all these transactions will be considered in a single block generation, so the verification cost will not increase with the number of transactions to be verified and included in a block. This is an added advantage of the scheme in the practical case of dense heterogeneous small cell system.

The signalling cost of the scheme can be considered in two different parts. In the first part, the source node will send the keys it is using for creating the tags as transactions to the small cell cloud. The small cell cloud will collect all such transactions over the specific time between two successive blocks and verify the trustworthiness of each transaction using the signature by the source. All the verified transactions are considered to be the part of new block mined using the algorithm and becomes a part of the blockchain. The time taken for this can be considered as the processing time of a block. In case of BigchainDB, for our simulations we considered a 1 second time gap between the formation of two blocks. Then any small cell can access to the verified blocks and it forms the second part of the signalling. Fig. 5 illustrates the signalling time for a block generation. Thus the total signalling cost of the system can be considered as the cost of sending each transaction to the cloud and retrieving the mined block from the cloud. It should be considered that if there are more than one small cell initiating the transactions during the collecting time, then all these transactions will be accumulated in a single block. This improves the number of connections each small cell need to make considerably. Even in the worst case scenario
of all small cells need to share keys with all other small cells, the total number of communications is two times the number of small cells compared to a $N \times (N-1)$ connections required in the conventional secure channel scenario. Fig. 4 depicts a comparison graph of the number of connections made in the total environment for key management in the two scenarios.

V. CONCLUSIONS

Network coding enabled mobile small cells can provide high resiliency and ensure efficient use of the available spectrum for wireless networks. It will also help to achieve low latency communication with minimal energy requirements. However, the security challenges, especially pollution attacks, can degrade the performance of the network coding enabled networks considerably. Existing integrity schemes in network coding needs to be extended to the small cell environment. To the best of our belief, this paper is one of the initial steps towards a homomorphic MAC based integrity scheme in a network coding enabled mobile small cells. One of the main challenges to implement MAC base integrity schemes for network coding is to ensure that every node will have enough keys to verify the MACs attached to the packets. With the highly unpredictable network of mobile small cells, it is practically impossible to share the keys beforehand. We propose a dynamic key management scheme enabled by blockchains. It ensures that whenever a node requires to transmit, it can share the keys with all the small cells around using the blockchain which enables any node in any small cell to verify the packets with corresponding tags stored in the blockchain. This will help to extend the efficient integrity schemes to the new small cell environment with least overhead. Even though this proposal is yet to be tested with the real small cell environment, it tries to resolve an issue which is rarely addressed before. From the analysis of the proposed approach with bigchainDB, employing a blockchain provides an efficient solution to the problem of key management in small cell environment than the conventional approach of creating secure channel between all the different cells.

REFERENCES

[1] V. Adat and B. Gupta, “Security in internet of things: issues, challenges, taxonomy, and architecture,” Telecommunication Systems, vol. 67, no. 3, pp. 423–441, 2018.
[2] C. Tselios and G. Tsolis, “On QoE-awareness through virtualized probes in 5G networks,” in 2016 IEEE 21st International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD), Oct 2016, pp. 159–164.
[3] G. Bianchi, E. Biton, N. Blefari-Melazzi, I. Borges, L. Chiavaviglio, P. Cruz Ramos, P. Eardley, F. Fontes, M. J. McGrath, L. Natarianni et al., “Superfluidity: a flexible functional architecture for 5G networks,” Transactions on Emerging Telecommunications Technologies, vol. 27, no. 9, pp. 1178–1186, 2016.
[4] L. Bolivar, C. Tselios, D. Mellado, and G. Tsolis, “On the deployment of an open-source, 5G-aware evaluation testbed,” in IEEE International Conference on Mobile Cloud Computing, Services and Engineering (Mobile Cloud), March 2018, pp. 1–6.
[5] C. Tselios and G. Tsolis, “A Survey on Software Tools and Architectures for Deploying Multimedia-Aware Cloud Applications,” in Lecture Notes in Computer Science: Algorithmic Aspects of Cloud Computing, Springer International Publishing, 2016, vol. 9511, pp. 168–180.
[6] S. Mumtaz, K. M. S. Huq, J. Rodriguez, S. Ghosh, E. E. Ugwuanyi, M. Iqbal, T. Dagkiulas, S. Stavroul, L. Karanis, I. Politis, A. Lykourgiotis, T. Chrysikos, P. Nakou, and P. Georgakopoulos, “Self-organization towards reduced cost and energy per bit for future emerging radio technologies - sonet,” in 2017 IEEE Globecom Workshops (GC Wkshps), Dec 2017, pp. 1–6.
[7] J. Rodriguez, A. Radwan, C. Barbosa, F. H. P. Fitzek, R. A. Abd-Alhameed, J. M. Noras, S. M. R. Jones, I. Politis, P. Galiotis, G. Schulte, A. Rayit, M. Sousa, R. Alheiro, X. Gelabert, and G. P. Koutroulis, “SECRET - Secure network coding for reduced energy next generation mobile small cells: A European Training Network in wireless communications and networking for 5G,” in 2017 Internet Technologies and Applications (ITA), Sept 2017, pp. 329–333.
[8] S. T. Basaran, G. K. Kurt, M. Uysal, and I. Altunbas, “A tutorial on network coded cooperation,” IEEE Communications Surveys & Tutorials, vol. 18, no. 4, pp. 2970–2990, 2016.
[9] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” IEEE Transactions on Information Theory, vol. 52, no. 10, pp. 4415–4430, Oct 2006.
[10] N. Cai and R. W. Yeung, “Secure network coding,” in Proceedings IEEE International Symposium on Information Theory, 2002, pp. 323–327.
[11] A. Antonopoulos and C. Verikoukis, “Cops: Cooperative statistical misbehavior mitigation in network-coding-aided wireless networks,” IEEE Transactions on Industrial Informatics, vol. 14, no. 4, pp. 1436–1446, April 2018.
[12] S. Agrawal and D. Boneh, “Homomorphic MACs: MAC-Based Integrity for Network Coding,” in Applied Cryptography and Network Security, M. Abdalla, D. Pointcheval, P.-A. Fouque, and D. Vergnaud, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 292–305.
[13] Y. Li, H. Yao, M. Chen, S. Jaggi, and A. Rosen, “Ripple authentication for network coding,” in INFOCOM, 2010 Proceedings IEEE, IEEE, 2010, pp. 1–9.
[14] P. Zhang, Y. Jiang, C. Lin, H. Yao, A. Wasef, and X. Shen, “Padding for orthogonality: Efficient subspace authentication for network coding,” in 2011 Proceedings IEEE INFOCOM, April 2011, pp. 1026–1034.
[15] A. Esfahani, G. Mantas, J. Rodriguez, and J. C. Neves, “An efficient homomorphic mac-based scheme against data and tag pollution attacks in network coding-enabled wireless networks,” International Journal of Information Security, vol. 16, no. 6, pp. 627–639, 2017.
[16] A. Esfahani, D. Yang, G. Mantas, A. Nascimento, and J. Rodriguez, “Dual-homomorphic message authentication code scheme for network coding-enabled wireless sensor networks,” International Journal of Distributed Sensor Networks, vol. 11, no. 7, p. 510251, 2015.
[17] V. Adat, I. Politis, C. Tselios, P. Galiotis, and S. Kotsopoulos, “On Blockchain Enhanced Secure Network Coding for 5G Deployments,” in 2018 IEEE Global Communications Conference (GLOBECOM), Dec 2018, pp. 1–7.
[18] V. Adat, I. Politis, C. Tselios, and S. Kotsopoulos, “Secure network coding for sdn-based mobile small cells,” in Broadband Communications, Networks, and Systems, V. Sucasas, G. Mantas, and S. Althumbat, Eds. Cham: Springer International Publishing, 2019, pp. 347–356.
[19] C. Tselios, I. Politis, and S. Kotsopoulos, “Enhancing SDN security for IoT-related deployments through blockchain,” in 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN), Nov 2017, pp. 303–308.
[20] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” 2008.
[21] G. Wood, “Ethereum: A secure decentralised generalised transaction ledger,” Ethereum project yellow paper, vol. 151, pp. 1–32, 2014.
[22] A. Lei, H. Cruickshank, Y. Cao, P. Asuquo, C. P. A. Ogah, and Z. Sun, “Blockchain-based dynamic key management for heterogeneous intelligent transportation systems,” IEEE Internet of Things Journal, vol. 4, no. 6, pp. 1832–1843, 2017.
[23] T. McConaghy, R. Marques, A. Müller, D. De Jonghe, T. McConaghy, G. McMullen, R. Henderson, S. Bellemare, and A. Grañzotto, “Bigchaindb: a scalable blockchain database,” white paper, BigChainDB, 2016.
[24] L. Walsh, V. Akhmechet, and M. Gluhkovsky, “Rethinkingdb-rethinking database storage,” 2009.
[25] V. Buterin, “Slasher: A punitive proof-of-stake algorithm,” Ethereum Blog URL: https://blog.ethereum.org/2014/01/15/slasher-a-punitive-proof-of-stake-algorithm, 2014.