Applying Diffie-Hellman Algorithm to Solve the Key Agreement Problem in Mobile Blockchain-based Sensing Applications

Nsikak Pius Owoh[1], Manmeet Mahinderjit Singh[2]
School of Computer Sciences, Universiti Sains Malaysia, USM 11800, Penang, Malaysia

***Abstract*—Mobile blockchain has achieved huge success with the integration of edge computing services. This concept, when applied in mobile crowd sensing, enables transfer of sensor data from blockchain clients to edge nodes. Edge nodes perform proof-of-work on sensor data from blockchain clients and append validated data to the chain. With this approach, blockchain can be performed pervasively. However, securing sensitive sensor data in a mobile blockchain (client/edge node architecture) becomes imperative. To this end, this paper proposes an integrated framework for mobile blockchain which ensures key agreement between clients and edge nodes using Elliptic Curve Diffie-Hellman algorithm. Also, the framework provides efficient encryption of sensor data using the Advanced Encryption Standard algorithm. Finally, key agreement processes in the framework were analyzed and results show that key pairing between the blockchain client and the edge node is a non-trivial process.**

Keywords*—Internet of Things; mobile crowd sensing; edge computing; sensor data encryption; mining; smart contract

I. INTRODUCTION

Mobile crowd sensing (MCS) has become an attractive method of gathering personal and environmental data [1]. It takes advantage of sensors (accelerometer, gyroscope, GPS, camera, etc.) and the communication capability of smart devices such as smartphones to collect and transmit large scale sensor data at low cost [2]. These sensors acquire useful data in several domains, including but not limited to environmental monitoring [3], healthcare [4], traffic monitoring [5]. Basically, a crowd sensing platform consists of a cloud-based system and a group of sensing devices (mobile users). The platform publishes a set of sensing task with various purposes, while the mobile users participate in the sensing task [6]. Mobile crowd sensing also plays a key role in the actualization of smart cities [7]. Cities are considered "smart" when they have among other things: intelligent versatility, smart administration, smart citizens, intelligent economics and intelligent life [8]. Most importantly, such cities should also be able to share data using information and communication technology (ICT) [9].

Despite the benefits of MCS, challenges such as incentivizing participants [10, 11], quality and reliability of sensed data [12], energy usage of mobile sensing devices [5], sensor data annotation [13], security and privacy [14, 15] still exist. Due to the sensitive information of users gathered and transmitted by sensing devices, different mechanisms have been proposed to ensure secure sensing in MCS applications [16]. Unfortunately, security and privacy still remain a pressing issue as an active attacker can intercept and modify transmitted sensor data (data in motion) from a mobile sensing device like the smartphone [17] and/or can alter stored data (data at rest).

Recently, the inherent attributes of blockchain technology have been harnessed to provide security and privacy in IoT [18] and specifically MCS applications [19]. The adoption of this technology for these purposes is not surprising, as its previous application in cryptocurrency has recorded some success. The gains of blockchain based cryptocurrency technology include; transformed payments, as middlemen are taken out of the loop and reduce merchant payment fees to below 1%, as well as the removal of delays, as users receive transferred funds instantly without having to wait for days [20]. Apart from its use in cryptocurrencies and smart contracts, blockchains have been applied in social services [21], smart living applications [22], supply chain management [23], intelligent transportation systems [24], data storage [25], identity management [26], smart cities [2].

Blockchain is a technology that reads, stores and validates transactions in a distributed database system [27]. The stored data can be cryptocurrency (Bitcoin) [28], a contract [29] or even personal data [30]. Another definition of blockchain is that, it is a security mechanism that ensures immutability, anonymity and auditability of electronic transactions [16]. It acts as a distributed ledger (a virtual book that stores previous transactions) allowing data to be shared among a network of peers by implementing a chain of timestamped blocks that are connected by cryptographic hashes. With this mechanism, untrustworthy participants (such as those in mobile crowd sensing) can reach a consensus and perform transactions without the involvement of third-parties. Blockchain can either be public (permissionless), private (permissioned), or consortium. Any user with internet access can join the network by taking part in block validation and smart contracts creation in public blockchains. Private blockchains on the other hand, controls users' right to validate block transactions and develop smart contracts [31]. Private blockchain offers privacy and efficiency of transactions. Consortium blockchains are somewhat private and grant selected nodes full access.

However, before using a blockchain, a peer-to-peer network with all interested nodes must be created. All

participating nodes are allocated a pair of private and public keys used for transactions [20]. When a transaction is performed by a node, it signs (using the private key) and broadcasts the transaction to the nearest peer. This ensures authentication as the transaction can only be signed by a user with the exact private key. Integrity is also maintained as data modification is easily detected on signed transactions. Received transactions are verified by miners and validated using consensus algorithms (such as proof-of-work, proof-of-stake, etc.), then added to the chain [32].

The much-needed security services in any system (confidentiality, privacy, integrity, availability and non-repudiation) are provided by blockchain. As regard confidentiality, this technology secures data (transaction details, account and wallet balances, assets, price, and smart contracts' business logic) from unauthorized persons (third parties) and can be achieved using encryption. Meanwhile, privacy secures the identity of blockchain users (participants) against disclosure and is actualized with the use of pseudonyms (64-bit addresses). On the other hand, integrity guarantees the immutability of transactions using cryptographic mechanisms such as hash functions and digital signature. Availability ensures that users of a certain system can use it at any time as such service is always available for legitimate users. Blockchain-based systems achieve this by implementing multiple connections with several users and ensuring that blocks are decentralized and replicated across the network. Non-repudiation ensures that an individual cannot deny any action performed in a system as evidence of actions such as money transfer, purchase authorization and sent messages are digitally signed. In blockchain, this security service is achieved using the Elliptic Curve Digital Signature (ECDSA). Blockchain implements mechanisms that ensure stringent integrity and availability of data. However, ensuring confidentiality has been difficult [20].

Despite all the advantages and uses of blockchain, this technology still faces some challenges. Firstly, scalability is a lingering problem owing to the proliferation of its usage and the increase in the number of daily transactions [33]. The block size in blockchain increases transaction latency especially in small transactions as preference is given to transactions with bigger transactional fees by miners. Thus, implementing blockchain in IoT becomes difficult since IoT applications deal with large sensor data that require high speed processing. However, redesigning of blockchains and storage optimization are proposed solutions to scalability issues in blockchain [34].

Secondly, security has always been an issue in open networks such as public blockchains and confidentiality which is a key security element is low in distributed systems such as this [20]. In addition, integrity which is an important function of blockchain offered by its immutability feature has a number of issues. Another strength of this technology is the duplication of data blocks to all nodes ensuring availability of data. Although this makes a single point of failure impossible, it is theoretically proposed that a 51% attack is still possible [20]. Privacy leakage is another weakness of blockchain as details and balances of all public keys can be seen by everyone in the network. This could lead to leakage of users' sensitive information when blockchain is employed in sensing applications. Mixing and anonymous techniques have been proposed as solutions to privacy leakage issues. Unfortunately, [35, 36] showed that de-anonymization is possible. Meanwhile, third-party is mostly needed with mixing techniques which introduces bottlenecks. Lastly, blockchains are faced with the issue of selfish mining, where a block is vulnerable to cheating when a minimal hashing power is used. In selfish mining, miners hoard mined blocks without broadcasting them to the network and generate a private branch which is broadcasted only when specific requirements are satisfied. This allows selfish miners to continue mining the private chain while honest miners waste their time and resources.

The use of blockchain as an underlying technology in IoT-based applications has gained acceptance both in the academia and industry. However, its practical use in mobile applications (mobile blockchain) has not been fully explored. A major reason being that, mining which requires high computational resources cannot be performed on resource-constrained mobile devices (such as smartphones) [21, 37]. In an effort to bridge this gap, edge computing approach is employed [33]. This integration allows mobile users to run the mobile blockchain application with the aid of edge computing nodes; serving as miners to mobile users and tagged by a service provider. Using this approach, blockchains can be implemented in mobile crowd sensing applications hence utilizing its full potentials on sensor data.

Edge computing supports blockchain and blockchainless Directed Acyclic Graph (DAG) applications [38] using one or more high-end computers (cloudlets) acting like a cloud [20]. These cloudlets respond swiftly to compute-intensive tasks requested by the node layer (bockchain node). Mobile blockchain together with edge computing has been used to improve social welfare [21]. Also, Xiong et al., Zhu et al. [37, 39] employed edge computing for mobile blockchain. These works focus on improving services rendered by the edge computing service providers such as enhancing pricing [40], or placement of mobile edge applications. However, secure data transmission between the mobile blockchain client (smartphones) and the edge nodes (miners) using effective key agreement and data encryption mechanisms have received little attention. Motivated by this, we propose a framework that secures sensor data transmitted between mobile and edge nodes during sensing activities. Different from the approach used in Conoscenti et al., Dorri et al., Zyskind and Nathan [18, 44, 49] where symmetric keys are transmitted with the encrypted data which makes sensor data susceptible to attack, we employ public key cryptography for key establishment between blockchain nodes.

The following contributions are made in this paper:

- We present a key agreement protocol using public key cryptography for secure key exchange between mobile blockchain clients and edge nodes in an MCS scenario.
- We present a technique to secure sensor data transmitted between mobile nodes and edge nodes (miners) using symmetric data encryption.

- We evaluate the proposed framework based on the computational time of the ECDH key agreement protocol and the execution time of the AES encryption scheme.

In this paper, we use the word "client" interchangeably with "mobile blockchain client". The rest of this paper is structured as follows: Section II presents a review of related works on blockchain-based security schemes for Internet of Things as well as frameworks for mobile blockchain. In Section III, we present the methodology and implementation of our proposed secure mobile blockchain framework for MCS. Further discussion on the results of the implementation of the proposed framework is presented in Section IV. We conclude the paper in Section V.

II. Related Works

Blockchain enhances the security of IoT devices for example in remote attestation which deals with the verification of trustworthiness of underlying Trusted Computer Base (TCB) [41]. Blockchain based systems do not depend on a specific central server or cloud due to the vulnerabilities that exist in traditional cloud-centered IoT architectures. For instance, the cloud being a single point of failure (due to attacks, maintenance and other software issues) [23, 42]. This section presents some works that employ blockchain to enhance security and privacy in IoT-based applications.

A. Blockchain-based Security and Privacy Preservation Schemes for IoT

To solve the problem of identity certification in IoT, where a provider in charge of authorizing entities can also block them, Kravitz and Cooper [43] proposed the use of permissioned blockchain for the management and security of IoT nodes. With the proposed system, asymmetric keys are rotated thereby offering security against attacks. To protect users' privacy, a blockchain-based decentralized personal data management system that maintains ownership of data by users was proposed in Zyskind and Nathan [44]. The system addresses privacy issues such as data ownership, data transparency and auditability. Similarly, a privacy-aware blockchain connected gateway was proposed for privacy preferences management in Cha et al. [45]. The blockchain gateway employs blockchain technology to ensure that user preferences are not modified thereby improving user privacy protection in legacy IoT devices. The owners of IoT devices, the blockchain gateway administrators and the end users are the three participants that can use the proposed blockchain gateway. The authors employed the Ethereum blockchain platform which allows the administrator to develop smart contract for the device as well as manage privacy policies.

In an effort to secure Electronic Health Records (EHR), Zitta et al. [46] employed smart contracts on an Ethereum blockchain to develop intelligent EHR that are stored in individual nodes. Garman et al. [47] proposed an anonymous credential authentication scheme that does not require a trusted credential issuer rather uses a public append-only ledger (blockchain). Using this system, privacy of users is preserved without the need for trusted third parties. Name value mappings are offered using Namecoin (a system developed on Bitcoin's Blockchain) for the storage of public keys with the associated credential.

A blockchian-based two-factor authentication scheme was proposed in Wu et al [48]. With this scheme, security of sensitive data is guaranteed through authentication and authorization. Furthermore, the proposed scheme uses two smart contracts: the device contract for the storage of device profiles and the relationship contract for the storage of associated device paring information. Authors evaluated the performance of the scheme by measuring the memory and CPU usage of individual nodes in the system. Privacy issues experienced when third party mobile services were employed were addressed in Zyskind and Nathan [44] using a blockchain-based application. For the application to function effectively, a set of permissions (location, list of contacts, camera, etc.) needs to be granted when initially signing up to any mobile application. Three entities including: mobile phone users, service providers and the nodes maintaining the blockchain make up the proposed decentralized system. Only two types of transactions are permitted in the proposed blockchain network; T_{access} for access control management and T_{data} for data storage and retrieval. The identity and corresponding permission of each user of a service is transmitted to the blockchain in a T_{access} transaction. Encryption is performed on data collected from the user's mobile phone and then stored off-chain while storing only the hashes of the data in the private blockchian. Data in a T_{data} transaction can be queried by the user and service.

Also, a blockchain-based smart home system was proposed in Dorri et al. [18]. The system consists of three tiers namely: smart home tier, overlay tier and the storage tier. The smart home tier includes as it core components transactions, home miner and the local storage. Confidentiality, integrity and availability are offered in the smart home tier. The system offers security against Distributed Denial of Service (DDOS) and linking attacks. Furthermore, packet overhead, time overhead and energy usage were metrics used to evaluate the performance of the proposed system. The authors concluded that encryption and hashing operations performed by the miner on all transactions are non-trivial processes compared to the encryption operation done by individual devices. The work in Conoscenti et al [49] uses blockchain technology in place of a centralized server for sensor data storage. Like cryptocurrencies, sensor data in the proposed system are managed by users via a distributed database. Also, symmetric keys are used for encryption of data to ensure data confidentiality. However, sending shared keys together with generated data for verification of data contents by miners undermines privacy and security.

To ensure privacy and confidentiality of shared data in blockchain-based IoT, Rahulamathavan et al [50] proposed the use of attribute-based encryption (ABE). Similar to the work in [51], the authors also employed a hierarchical IoT network method that dedicates a cluster head for certain set of IoT sensors. The cluster head in this case has resources enough to carry out data processing and encryption. Proposed works in Conoscenti et al. [49], Dorri et al. [18], and Zyskind and Nathan [44] all focused on the use of symmetric encryption where shared keys are transmitted with encrypted

data to participants. Unfortunately, an eavesdropper can decrypt encrypted data using the captured keys when such encryption schemes are employed. However, a countermeasure is the use of public key cryptography for key agreement between participants.

B. Blockchain-based Access Control Schemes for IoT

Controlling access to IoT resources can be achieved using blockchain technology. For instance, a blockchain-based scheme that controls access to medical records was proposed in Xia et al. [52]. The scalable system grants legitimate users access to Electronic Health Record (EHR), from a pool of shared data, after performing identity and cryptographic key verification using a permissioned blockchain. The proposed system functions well in areas where conventional access control approaches like firewalls, passwords and intrusion detection systems fail. Similarly, in Ouaddah et al [53], access to IoT applications is controlled via a proposed FairAccess system that hybridizes the Bitcoin blockchain and the Ethereum smart contracts technology. The system provides access control management in an IoT-based environment.

To ensure end-to-end (E2E) security for IoT data in motion, Vučinić et al. [54] proposed the Object Security Architecture (OSCAR) for the IoT. The important issues with the Datagram Transport Layer Security (DTLS) protocol were addressed in the proposed architecture by securing the payload at the application layer using blockchain. Using this model, resource servers can either store their resources locally or on a proxy server after encrypting and signing them. On the other hand, Alphand et al. [55] proposed an End-to-End scheme that ensures authorized access to IoT resources by hybridizing OSCAR [54] and ACE (Authentication and Authorization for Constrained Environments) frameworks. The authors employed a trustless authorization blockchain in place of the single trusted authorization server in the ACE framework. This enhances the ACE authorization model as resource access control becomes secure, and flexible.

Another blockchain-based architecture for access control of IoT devices was proposed in Pinno et al. [56]. The authors claim that the proposed architecture which is decentralized and transparent, solves the FairAccess problem associated with traditional architectures and can be integrated with several IoT access control models. To ensure distributed and trustworthy access control for IoT, Zhang et al. [57] employed smart contract-based blockchain technology that consists of several access control contracts (ACCs), a single judge contract (JC) and one register contract (RC). In the proposed framework, managing data records is the main goal of the smart contracts.

C. Mobile Edge Node Blockchain

Recently, blockchain has been implemented in mobile applications using the edge computing concept [21]. Some Android applications such as Easy Miner [58], LTC and Scrypt Miner PRO [59] have been developed for performing mining operations on mobile devices. However, they currently lack full implementation. On the other hand, a novel mobile-commerce application called MobiChain which employs blockchain technology for secure transactions was presented in Suankaewmanee et al. [32]. The authors developed a Mobile Blockchain Application Programming Interface (MBAPI) for effective mining operations on mobile devices. Computation time, energy consumption, and memory utilization were metrics used to evaluate the performance of the proposed module.

Edge computing for mobile blockchain was introduced in [40]. In the presented prototype, IoT or mobile devices (Android devices) carry out mining on an edge computing server. The nodes (using Ethereum) serve as miners that install mobile client applications. Internal sensors such as accelerometer and GPS are used to record data (transactions) of mobile peer-to-peer communication by the application. In Jiao et al. [21], the authors also employed edge computing services for mobile blockchain applications and proposed an auction-based market model which comprises of the blockchain owner, edge computing service provider (ESP) and miners. The proposed model enhances social welfare and simulation results show the efficiency of the proposed model in solving the social welfare maximization problem.

Edge computing has made it possible for mobile blockchain to reach its full potentials, as edge nodes (miners) supported with edge computing service provider (ESP) can solve the PoW puzzle offloaded by the mobile blockchain client. The edge computing concept makes it practical to employ blockchain in mobile crowd sensing applications that deal with large chunks of sensor data from numerous sensing devices. However, the security of sensed data offloaded to edge nodes from blockchain clients remains a major challenge, which this paper aims to solve.

III. Methodology

There are four phases in the proposed Mobile Blockchain Security Framework (MBCSF) as shown in Fig. 1. In what follows, we provide details of each phase.

A. Key Agreement

This is the first phase of the framework where a variant of Elliptic Curve cryptography (i.e. EDCH) algorithm is employed for key establishment between the blockchain client (smartphone) and miners. First, we provide a brief description of the Elliptic Curve Cryptography (ECC) and its Elliptic Curve Diffie Hellman (ECDH) variant.

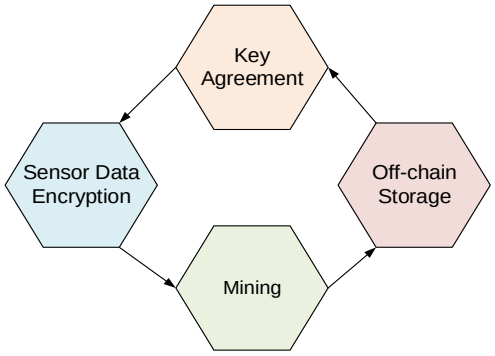

Fig. 1. The Proposed Mobile Blockchain Security Framework (MBCSF).

1) Elliptic Curve Cryptography (ECC): Elliptic Curve Cryptography (ECC) is one of the public key cryptography (PKC) primitives, which is based on discrete logarithm (DL) [60]. Its popularity stems from its small key size and low computational overhead, which justifies why it is appropriate for mobile devices and delay-sensitive applications such as mobile crowd sensing. For instance, a 160 bit key in ECC provides equivalent security as a 1024 bit key in RSA [61]. ECC's mathematical operations are defined over the elliptic curve as shown in (1):

$$y^2 \equiv x^3 + ax + b \bmod p \quad (1)$$

where $4a^3 + 27b^2 \neq 0 \bmod p$. Each generated value of *'a'* and '*b'* produces a unique elliptic curve. All points *(x, y)* which satisfy the equation above and a point at infinity lies on the elliptic curve [61]. The private key is a randomly generated number while the public key is a point on the elliptic curve. The multiplication of the private key with the primitive element (generator) *P* generates the public key. The primitive element *P*, the curve parameters *'a'* and '*b'* make up the domain parameter of ECC [62].

2) Elliptic Curve Diffie Hellman (ECDH): ECDH is a key agreement protocol that allows two communicating entities establish a shared secret key. This ensures exchange of public information between both parties (using parameters). The available public data and their respective private data are used to compute the shared secret. This ensures that an attacker (such as an eavesdropper) without knowledge of the private keys of each party, cannot compute the shared secret from the available public information. Using ECDH, a shared secret between two communicating parties (A and B) can only be generated after both parties agree on Elliptic Curve domain parameters. With this in mind, we integrate ECDH algorithm in our proposed framework. The major aim of our proposed framework is to eliminate the key distribution problem between the blockchain client and edge node. Descriptions of mathematical notations used in this paper are presented in Table I.

TABLE I. MATHEMATICAL NOTATIONS

Notations	Description
A *B* S_A	Blockchain client (smartphone) Blockchain miner Public key of the blockchain client
S_B	Public key of the blockchain miner
T_A	Private key of the blockchain client
T_B	Private key of the blockchain miner
P	Primitive element
E *X, Y*	Elliptic Curve Elliptic curve coordinates
K_{AB} P_T	Shared secret key Plaintext sensor data
C	Ciphertext

To securely establish keys, both the blockchain client and edge node (miner) must generate their private keys as shown in (2) and (3) from agreed domain parameters *(E, P)* using a 96-bits key generator.

$$T_A = K_{pr,A} \in \{2,3,\#E\}s \quad (2)$$

$$T_B = K_{pr,B} \in \{2,3,\#E\} \quad (3)$$

Thereafter, the blockchain client computes (4) as its public key:

$$S_A = K_{prA} = S_A.P \quad (4)$$

and sends S_A, *E*, *P* (public key, and domain parameters) to the miner.

The miner on the other hand, computes (5) as its public key:

$$S_B = K_{prA} = S_B.P \quad (5)$$

and sends S_B, *E*, *P* to the blockchain client. Both S_A and S_B are points on the elliptic curve and are computed using the point multiplication. With S_A and S_B, both the blockchain client and the miner compute (6) and (7) as joint secret:

$$K_{AB} = T_A.S_B = (X_{AB}.Y_{AB}) \quad (6)$$

$$K_{AB} = T_B.SA = (X_{AB}.Y_{AB}) \quad (7)$$

Fig. 2 illustrates the key agreement process between the blockchain client and the miner node.

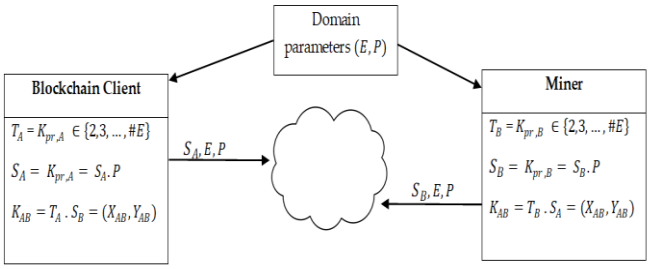

Fig. 2. Key Agreement Process in Proposed MBCSF using Elliptic Curve Diffie-Hellman.

B. Sensor Data Encryption

Encryption of sensor data only happens after a shared secret is established between the blockchain client and edge (miner) node. Encrypting large chunks of data such as those obtained from mobile sensors (blockchain client) with public key cryptography can be non-trivial even with ECC. Thus, a symmetric primitive such as AES is preferred as it requires lesser computations compared to ECC and RSA. To this effect, we employed AES with a 128 key bit length for encryption of sensor data in our proposed framework. Nevertheless, the shared secret K_{AB} from the key agreement phase is used to derive a session key which serves as the AES encryption key. Since only 128 key bit length can be used with AES, the last 32 bits of the 160 bit key of ECDH were dropped; using only the first 128 bits. Using this key, the blockchain client (smartphone) performs encryption on data from GPS, accelerometer and gyroscope sensors as shown in Fig. 3.

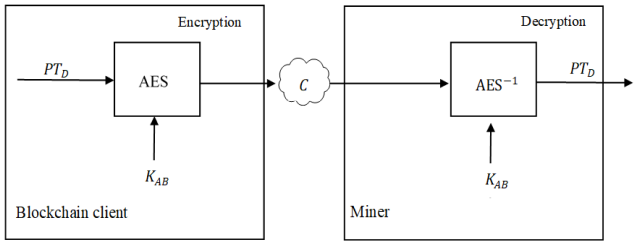

Fig. 3. Encryption and Decryption of Sensor Data in Proposed MBCSF using Advanced Encryption Standard.

Encrypted data are queued in a pool of unprocessed data awaiting validation by miners. Miners with the joint secret can derive the session key which is used to decrypt sensor data and commence the mining process. The algorithm for the encryption process is presented in Table II.

TABLE II. ALGORITHM TO ENCRYPT SENSOR DATA ON BLOCKCHAIN CLIENT (SMARTPHONE)

Algorithm 1: Encrypt Sensor data using AES-CBC
Input: Sensor data from blockchain client
1. Initialize *IV* 2. $CT_1 = Enc K_{AB}(PT_1) \oplus IV$ 3. $CT_n = Enc K_{AB}(PT_n) \oplus CT_{n-1}$ for $n \geq 2$ 4. Repeat for all *n*
Output: Encrypted sensor data to miner node

C. Mining

Mining and blockchain updates are performed at this stage of the framework. Integrity and validity in blockchains are ensured using the compute-intensive process referred to as mining [40]. To add a new block of data to existing blockchain, a miner has to solve a proof of work (PoW) to get a hash value that links the preceding block to the current block. On completion of the PoW, a broadcast of the result is made to other miners in the network for validation. The new block is then appended to the blockchain only when a consensus is reached by majority of the miners. Thereafter, a reward is given to the miner who successfully solved the PoW. The entire mining process (PoW) requires computational power which makes it difficult for resource-constrained devices such as smartphones to either take part in the mining or consensus process. Based on this fact, we adopted the Mobile Edge Computing (MEC) architecture [63] in our framework, which enables the deployment of local data centers and servers by an edge computing service provider (ESP) at the "edge" of mobile networks. The ESP in our framework supports offloading of proof-of-work puzzle by blockchain clients (smartphones) using the uniform pricing scheme. It also handles data storage and request distribution from several end users.

From the previous stage where data were encrypted and offloaded to the edge node, the miner willing to solve the PoW first decrypts the sensor data using the derived session key from the shared secret computed in (6). On successful mining and validation of sensor data, the hash value of the data block is stored in the blockchain while the sensor data is encrypted again by the miner and sent to the off-chain storage. Furthermore, we ensured that each hashed value points to the respective encrypted data in the off-chain storage. Indeed, the approach is preferred since large amount of sensor data from mobile crowd sensing users cannot be stored in the blockchain. More so, MCS stakeholders might need to reuse certain sensor data for decision making purposes.

With this framework, transaction details (sensor readings) in smart contracts are not transmitted in plaintext and this secures sensitive information from eavesdroppers. Also, the proposed framework ensures that sensor readings (transaction details) cannot be gathered or analysed with "off-chain" metadata to disclose any information about participants. This therefore maintains the confidentiality of sensitive information such as location details of users.

D. Off-Chain Storage

The data storage stores encrypted sensor data containing location information of MCS users. These data can be queried by users when sensing information need to be shared among sensing participants and blockchain members. We integrated cloud storage to achieve larger storage space while maintaining access control mechanisms on stored data. The proposed framework provides a cloud storage option where users' profile and environmental data from peers such as servers, smartphone sensors can be stored.

The proposed framework was implemented both on the blockchain client and the edge node (miner). For the blockchain client, we used a Samsung Galaxy S4 (GT19500) smartphone running Android version 5.0.1. Table III summarizes hardware and software features of the mobile and miner node used in our experiment. Android studio was used for implementation of the mobile blockchain client. The application gathers data from GPS, accelerometer and gyroscope sensors. On the client-side, we implemented the ECDH from the Spongy castle library using the JCE (Java Cryptography Extension).

We initialized the *KeyAgreement* class with the blockchain client's (smartphone) private key and the public key of the miner and then obtained the shared secret bytes by calling the *generateSecret()* function. The standard EC curve was used in our implementation. The generated keys are stored externally, and the key exchange performed successfully. With the secret key, sensor data offloaded for mining was encrypted using AES-CBC mode.

TABLE III. IMPLEMENTATION ENVIRONMENT FOR BLOCKCHAIN CLIENT AND EDGE NODE

Samsung Galaxy S4 (client)	Android 5.0.1
	2GB RAM, 16/32 GB storage
	Quad-core 2.3 GHz Krait 400 CPU, Adreno 330 GPU
Intel Xeon E5-2650	Ubuntu 16.04 LTS
	32 B DDR4 RAM, 1 TB Storage
	8 CPU cores, 16 threads

For the edge computing server, we employed the Intel Xeon E5-2650V4. The edge computing server is connected to the mobile device via a gateway (network hub). Ethereum smart contract was implemented using the solidity scripting language in a private blockchain network. We used the web3.js [64] (the official Ethereum Javascript API) for the object side to communicate with the matching geth client through HTTP connections. ECDH, ECDSA and AES algorithms run on the miner node as well. The shared secret generated at the point of connection with the mobile node is used for the decryption of transaction (sensor data) before mining is performed. This key is also used to encrypt sensor data after block validation by other miners.

IV. Results and Discussion

Architecturally, the proposed framework consists of three layers: the user layer, the management layer and the storage layer as shown in Fig. 4. Smart devices (such as smartphones and tablets) carried by users are classified under the user layer. Forming a peer-to-peer network, smart devices (referred to as blockchain clients) are connected to the blockchain via the Ethereum smart contract. In this layer, sensor data are acquired from users and the environment. The management layer on the other hand offers data distribution and decentralization to other layers in the framework. Edge node computing and security are some of the services offered in this layer.

The storage layer serves as an off-chain storage for validated (mined) sensor data. The off-chain solution is chosen in our framework to avoid challenges that exist with storing large amount of data in the blockchain, especially when dealing sensor data from numerous devices in MCS [65].

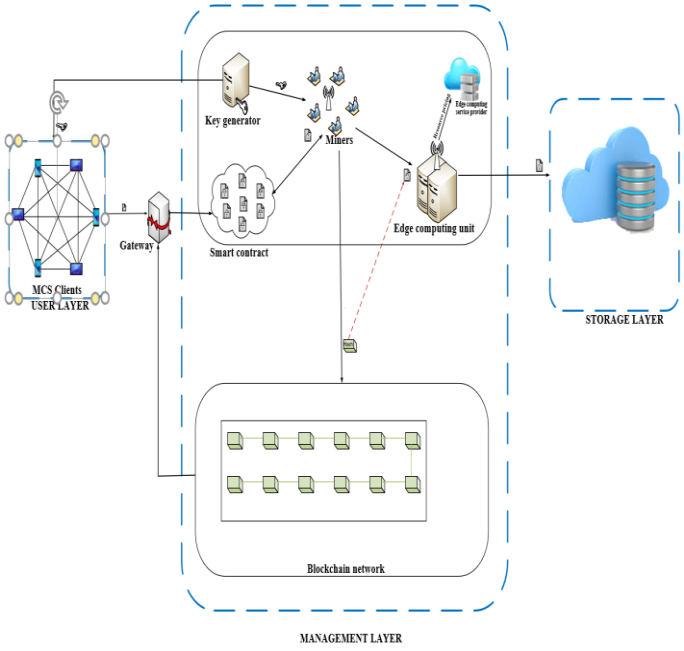

Fig. 4. Architecture of Proposed Mobile Blockchain-based Security Framework (MBCSF).

Blockchain technology has played a major role in the success of cryptocurrency. Recently, blockchains together with smart contracts has been implemented in other areas such as IoT, supply chain, healthcare, mobile crowd sensing, etc. However, its adoption in mobile crowd sensing is still in its early stage. Immutability, auditability, transparency and anonymity are some of the characteristics of blockchain. These features make it possible for blockchain technology to ensure security services such as data integrity, availability, non-repudiation and confidentiality. In reality, confidentiality in blockchain is only provided via anonymity, where addresses are used for transactions.

Applying blockchain in mobile crowd sensing applications implements the mobile blockchain concept where mobile users perform sensing activities, mining and block validation pervasively. However, since mining and block validation are non-trivial activities, we implemented our framework using edge computing. Mobile crowd sensing users can gather sensor data from their smartphones (mobile nodes) and offload encrypted data to the edge node. The edge node (miner) uses the shared secret computed using ECDH to decrypt sensor data and perform PoW. Data confidentiality is ensured as only the miners with the shared secret can decrypt encrypted data. Meanwhile, edge computing service providers cannot decrypt transactions which maintains security at the edge computing sub layer. Encryption using symmetric keys as employed in Conoscenti et al.; Dorri et al.; Zyskind and Nathan [18, 44, 49] does not guarantee effective data confidentiality, since keys are transmitted together with sensor data. An eavesdropper who listens to traffic between communicating parties can successfully capture encryption keys hence decrypt data meant for either party. Consequently, sensitive information of users are disclosed with the success of such an attack.

The need for an effective key agreement and distribution mechanism cannot be overemphasized especially when mobile blockchain is adopted for crowd sensing applications. Owing to the fact that mobile crowd sensing applications such as smart city applications gather sensitive information of users e.g. location data from GPS sensor, the robust key agreement protocol employed in the proposed framework ensures that the shared secret between communicating parties cannot be brute-forced by an attacker.

One major function of the proposed framework is the generation and distribution of secret keys between client and miner nodes in the blockchain. This process is implemented in the key agreement phase of the framework. Using the Elliptic Curve Diffie-Hellman algorithm, the framework ensures that only communicating nodes in the blockchain at any given time can compute the joint secret (session key) from their respective private keys. Sensor data are then encrypted using the computed keys. Employing encryption keys from a secure shared secret enhances the security of sensor data. Even when communication between the blockchain client and edge node is performed through a wireless channel which is susceptible to attacks, the proposed framework secures sensor data in mobile blockchain from attacks such as information disclosure and false data injection.

A. Security Analysis and Performance Evaluation

A proof of correctness of the key agreement algorithm (ECDH) adopted in the framework is given as follows:

The blockchain client computes its private key as (8), which is derived from (6):

$$T_A.S_B = S_A(T_BP) \quad (8)$$

At the same time, the miner node computes its (9) which is the private key obtained from (7):

$$T_B.S_A = S_B(T_AP) \quad (9)$$

As a result of this, both the blockchain client and the miner node compute similar keys represented in (10):

$$K_{AB} = T_B.S_A \quad (10)$$

The private keys of the blockchain client and the miner node T_A and T_B respectively are large integers used to generate their associated public keys S_A and S_B. Resultantly, only the two communicating parties with the matching private keys can compute the shared secret, hence the session key. That way, a secure key is established between blockchain nodes even when an unsecure channel is used. Consequently, when the encryption key is derived from the shared secret, sensor data are protected against information leakage. The proposed framework offers the following security services:

1) Confidentiality: Symmetric encryption of sensor data from blockchain clients (smartphones) to edge (miners) nodes guarantees data confidentiality in the proposed framework. Sensitive information of users are protected from eavesdroppers as only ciphertext messages are transmitted between nodes.

2) Integrity: This is a fundamental security service provided by blockchain technology as they are designed to store immutable information. In our proposed framework, hashes of sensor data are stored in the blockchain, making it difficult to modify any data content that have been validated and added to the chain.

3) Non-repudiation: Using the Ethereum smart contract, all transactions are digitally signed using the ECDSA algorithm. The proposed framework in this paper ensures that sensor data are signed by the sending device.

To evaluate the performance of the proposed framework, we calculated the computational cost using a Java program to obtain the running time in milliseconds (ms) of the key agreement component of the framework. Table IV and Fig. 5 shows that, key pairing between the blockchain client (smartphone) and the miner node takes longer time than other key agreement processes.

Also, we evaluated the execution time of the AES algorithm on both blockchain client and (edge) miner node. From the presented results in Fig. 6, encryption of sensor data is faster on the edge node when compared to blockchain client (smartphone). The high computing power of the edge node justifies this result.

TABLE IV. COMPUTATIONAL COST OF SECURITY COMPONENTS IN THE PROPOSED (MBCSF)

Security Component	Computational cost
Private key generator (96 bits)	0.5ms
ECDH pairing (160 bits)	95ms
Secret key generation (160bits)	0.9ms
EC point multiplication (160 bits)	1ms
EC point addition (160 bits)	0.8ms

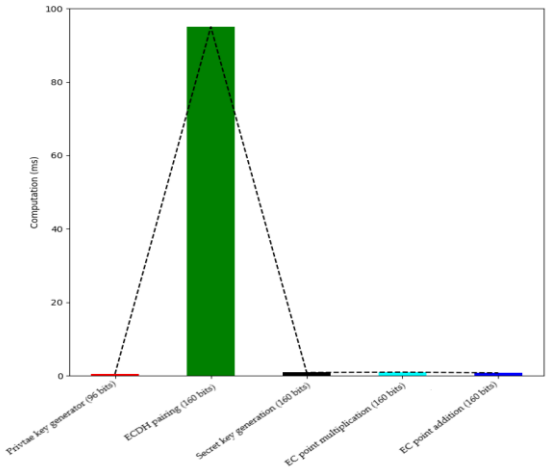

Fig. 5. Computational Cost of Security Components in the Proposed MBCSF.

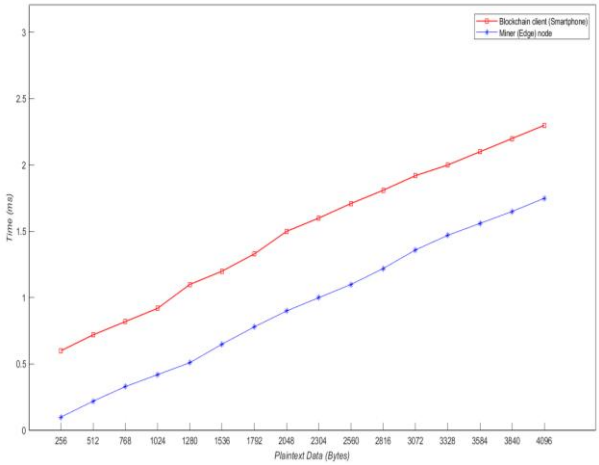

Fig. 6. Encryption Process in Blockchain Client and Miner Node.

V. CONCLUSION

In this paper, a security framework for mobile blockchain was presented. The framework designed for mobile crowd sensing applications consists of five steps (key agreement, sensor data encryption, mining and off-chain storage). With the framework, key agreement is achieved between mobile blockchain clients (smartphones) and edge (miner) nodes. Using ECDH as the key agreement algorithm, both communicating parties (blockchain client and miner node) employ public key cryptography for key generation. Adopting

elliptic curve cryptography enables the use of smaller key sizes to obtain maximum security level using a shared secret. The shared secret was used to derive the session key for the encryption of sensor data. Encrypted data in the proposed framework is secure as an attacker cannot obtain the private keys of the communicating parties.

REFERENCES

[1] Capponi, A., et al., A cost-effective distributed framework for data collection in cloud-based mobile crowd sensing architectures. IEEE Transactions on Sustainable Computing, 2017. 2(1): p. pp. 3-16.

[2] Biswas, K. and V. Muthukkumarasamy. Securing smart cities using blockchain technology. in High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), 2016 IEEE 18th International Conference on. 2016. IEEE.

[3] Cardone, G., et al., Fostering ParticipAction in Smart Cities: A Geo-Social Crowdsensing Platform,. IEEE Communications Magazine, Vol. 48, No. 14, pp.32–39., 2013.

[4] Khan, W.Z., et al., Mobile Phone Sensing Systems: A Survey. Ieee Communications Surveys & Tutorials, Vol. 15, No. 1, pp. 402-427., 2013.

[5] Ganti, R.K., F. Ye, and H. Lei, Mobile crowdsensing: Current state and future challenges. IEEE Commun. Mag., Vol. 49, No. 11, pp. 32–39., 2011.

[6] Nie, J., et al., A Stackelberg Game Approach Towards Socially-Aware Incentive Mechanisms for Mobile Crowdsensing. arXiv preprint arXiv:1807.08412, 2018: p. pp. 1-30.

[7] Cardone, G., et al., The participact mobile crowd sensing living lab: The testbed for smart cities. IEEE Communications Magazine, 2014. 52(10): p. pp.78-85.

[8] Gori, P., P.L. Parcu, and M.L. Stasi, Smart cities and sharing economy. 2015.

[9] Sharma, P.K., S.Y. Moon, and J.H. Park, Block-VN: A distributed blockchain based vehicular network architecture in smart City. Journal of Information Processing Systems, 2017. 13(1): p. pp.84.

[10] Wen, Y., et al., Quality-driven auction-based incentive mechanism for mobile crowd sensing. IEEE Transactions on Vehicular Technology, 2015. 64(9): p. pp.4203-4214.

[11] Jin, H., et al. Inception: Incentivizing privacy-preserving data aggregation for mobile crowd sensing systems. in Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing. 2016. ACM.

[12] Talasila, M., R. Curtmola, and C. Borcea, Mobile crowd sensing. Google Scholar, 2015: p. pp.

[13] Pius Owoh, N., M. Mahinderjit Singh, and Z.F. Zaaba, Automatic Annotation of Unlabeled Data from Smartphone-Based Motion and Location Sensors. Sensors, 2018. 18(7): p. pp.2134.

[14] He, D., S. Chan, and M. Guizani, User privacy and data trustworthiness in mobile crowd sensing. IEEE Wireless Communications, 2015. 22(1): p. pp.28-34.

[15] Zhang, D., et al., 4W1H in mobile crowd sensing. IEEE Communications Magazine, 2014. 52(8): p. pp.42-48.

[16] Jesus, E.F., et al., A Survey of How to Use Blockchain to Secure Internet of Things and the Stalker Attack. Security and Communication Networks, 2018. 2018: p. 1-28.

[17] Owoh, N.P. and M.M. Singh, Security analysis of mobile crowd sensing applications. Applied Computing and Informatics, 2018: p. pp. 1-11.

[18] Dorri, A., et al. Blockchain for IoT security and privacy: The case study of a smart home. in Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on. 2017. IEEE.

[19] Tanas, C., S. Delgado-Segura, and J. Herrera-Joancomartí, An Integrated Reward and Reputation Mechanism for MCS Preserving Users' Privacy, in Data Privacy Management, and Security Assurance. 2015, Springer. p. pp.83-99.

[20] Fernández-Caramés, T.M. and P. Fraga-Lamas, A Review on the Use of Blockchain for the Internet of Things. IEEE Access, 2018.

[21] Jiao, Y., et al., Social welfare maximization auction in edge computing resource allocation for mobile blockchain. arXiv preprint arXiv:1710.10595, 2017: p. pp. 1-6.

[22] Han, D., H. Kim, and J. Jang. Blockchain based smart door lock system. in Information and Communication Technology Convergence (ICTC), 2017 International Conference on. 2017. IEEE.

[23] Kshetri, N., Can blockchain strengthen the internet of things? IT Professional, 2017. 19(4): p. pp. 68-72.

[24] Lei, A., et al., Blockchain-based dynamic key management for heterogeneous intelligent transportation systems. IEEE Internet of Things Journal, 2017. 4(6): p. pp.1832-1843.

[25] Ateniese, G., et al. Accountable storage. in International Conference on Applied Cryptography and Network Security. 2017. Springer.

[26] Wilson, D. and G. Ateniese. From pretty good to great: Enhancing PGP using bitcoin and the blockchain. in International conference on network and system security. 2015. Springer.

[27] Bozic, N., G. Pujolle, and S. Secci. A tutorial on blockchain and applications to secure network control-planes. in Smart Cloud Networks & Systems (SCNS). 2016. IEEE.

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

[29] Wood, G., Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper, 2014. 151: p. pp. 1-32.

[30] Yue, X., et al., Healthcare data gateways: found healthcare intelligence on blockchain with novel privacy risk control. Journal of medical systems, 2016. 40(10): p. 218.

[31] Han, M., et al. Privacy reserved influence maximization in gps-enabled cyber-physical and online social networks. in Big Data and Cloud Computing (BDCloud), Social Computing and Networking (SocialCom), Sustainable Computing and Communications (SustainCom)(BDCloud-SocialCom-SustainCom), 2016 IEEE International Conferences on. 2016. IEEE.

[32] Suankaewmanee, K., et al. Performance analysis and application of mobile blockchain. in 2018 International Conference on Computing, Networking and Communications (ICNC). 2018. IEEE.

[33] Joshi, A.P., M. Han, and Y. Wang, A survey on security and privacy issues of blockchain technology. Mathematical Foundations of Computing, 2018. 1(2): p. pp. 121-147.

[34] Zheng, Z., et al. An overview of blockchain technology: Architecture, consensus, and future trends. in Big Data (BigData Congress), 2017 IEEE International Congress on. 2017. IEEE.

[35] Narayanan, A. and V. Shmatikov, How to break anonymity of the netflix prize dataset. arXiv preprint cs/0610105, 2006.

[36] De Montjoye, Y.-A., et al., Unique in the crowd: The privacy bounds of human mobility. Scientific reports, 2013. 3: p. pp.1376.

[37] Xiong, Z., et al. Optimal pricing-based edge computing resource management in mobile blockchain. in 2018 IEEE International Conference on Communications (ICC). 2018. IEEE.

[38] Yeow, K., et al., Decentralized consensus for edge-centric internet of things: A review, taxonomy, and research issues. IEEE Access, 2018. 6: p. 1513-1524.

[39] Zhu, H., C. Huang, and J. Zhou, EdgeChain: Blockchain-based Multi-vendor Mobile Edge Application Placement. arXiv preprint arXiv:1801.04035, 2018: p. pp. 1-9.

[40] Xiong, Z., et al., When mobile blockchain meets edge computing. IEEE Communications Magazine, 2018. 56(8): p. pp. 33-39.

[41] Park, J. and K. Kim. TM-Coin: Trustworthy management of TCB measurements in IoT. in Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on. 2017. IEEE.

[42] Li, X., et al. An IoT Data Communication Framework for Authenticity and Integrity. in Internet-of-Things Design and Implementation (IoTDI), 2017 IEEE/ACM Second International Conference on. 2017. IEEE.

[43] Kravitz, D.W. and J. Cooper. Securing user identity and transactions symbiotically: IoT meets blockchain. in Global Internet of Things Summit (GIoTS), 2017. 2017. IEEE.

[44] Zyskind, G. and O. Nathan. Decentralizing privacy: Using blockchain to protect personal data. in Security and Privacy Workshops (SPW), 2015 IEEE. 2015. IEEE.

[45] Cha, S.-C., et al., A Blockchain Connected Gateway for BLE-Based Devices in the Internet of Things. IEEE Access, 2018. 6: p. pp.24639-24649.

[46] Zitta, T., M. Neruda, and L. Vojtech. The security of RFID readers with IDS/IPS solution using Raspberry Pi. in Carpathian Control Conference (ICCC), 2017 18th International. 2017. IEEE.

[47] Garman, C., M. Green, and I. Miers. Decentralized Anonymous Credentials. in NDSS. 2014. Citeseer.

[48] Wu, L., et al. An out-of-band authentication scheme for internet of things using blockchain technology. in 2018 International Conference on Computing, Networking and Communications (ICNC). 2018. IEEE.

[49] Conoscenti, M., A. Vetro, and J.C. De Martin. Peer to Peer for Privacy and Decentralization in the Internet of Things. in Proceedings of the 39th International Conference on Software Engineering Companion. 2017. IEEE Press.

[50] Rahulamathavan, Y., et al. Privacy-preserving blockchain based IoT ecosystem using attribute-based encryption. in 2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS). 2017. IEEE.

[51] Dorri, A., S.S. Kanhere, and R. Jurdak. Smart city architecture and its applications based on IoT. in Proceedings of the Second International Conference on Internet-of-Things Design and Implementation. 2017. ACM.

[52] Xia, Q., et al., BBDS: Blockchain-based data sharing for electronic medical records in cloud environments. Information, 2017. 8(2): p. pp. 44.

[53] Ouaddah, A., A.A. Elkalam, and A.A. Ouahman, Towards a novel privacy-preserving access control model based on blockchain technology in IoT, in Europe and MENA Cooperation Advances in Information and Communication Technologies. 2017, Springer. p. pp. 523-533.

[54] Vučinić, M., et al., OSCAR: Object security architecture for the Internet of Things. Ad Hoc Networks, 2015. 32: p. pp. 3-16.

[55] Alphand, O., et al. IoTChain: A blockchain security architecture for the Internet of Things. in Wireless Communications and Networking Conference (WCNC), 2018 IEEE. 2018. IEEE.

[56] Pinno, O.J.A., A.R.A. Gregio, and L.C. De Bona. ControlChain: Blockchain as a Central Enabler for Access Control Authorizations in the IoT. in GLOBECOM 2017-2017 IEEE Global Communications Conference. 2017. IEEE.

[57] Zhang, Y., et al., Smart Contract-Based Access Control for the Internet of Things. arXiv preprint arXiv:1802.04410, 2018.

[58] EasyMiner, Easy Miner, Available at: https://play.google.com/store/apps/ details?id=com.mr.app.ui&hl=en. 2017.

[59] LTC, LTC and Scrypt Miner PRO, Available at: https://play.google. com/store/apps/details?id=com.miner.scrypt&hl=en. 2017.

[60] Paar, C. and J. Pelzl, Understanding cryptography: a textbook for students and practitioners. 2009: Springer Science & Business Media.

[61] Lee, Y.S., E. Alasaarela, and H. Lee. Secure key management scheme based on ECC algorithm for patient's medical information in healthcare system. in Information Networking (ICOIN), 2014 International Conference on. 2014. IEEE.

[62] Shou, Y., H. Guyennet, and M. Lehsaini. Parallel scalar multiplication on elliptic curves in wireless sensor networks. in International Conference on Distributed Computing and Networking. 2013. Springer.

[63] Abbas, N., et al., Mobile edge computing: A survey. IEEE Internet of Things Journal, 2018. 5(1): p. pp. 450-465.

[64] Javascript, Web3 javascript api to interact with ethereum nodes. [Online].Available:https://github.com/ethereum/wiki/wiki/JavaScript-API. 2018.

[65] Lazarovich, A., Invisible Ink: blockchain for data privacy. 2015, Massachusetts Institute of Technology.