Subscription Management for Beyond 5G and 6G Cellular Networks using Blockchain Technology

[1, 2] Nischal Aryal, [2] Fariba Ghaffari, [1, 2] Emmanuel Bertin, [2] Noel Crespi
[1] *Orange Innovation, 14000 Caen, France*
[2] *SAMOVAR, Telecom SudParis, Institut Polytechnique de Paris, 91120 Palaiseau, France*
{nischal.aryal, emmanuel.bertin}@orange.com, fariba.ghaffari@telecom-sudparis.eu, and noel.crespi@it-sudparis.eu

***Abstract*—As Mobile Network Operators (MNOs) prepare to interconnect diverse technologies to existing cellular networks, it is critical to assess the capabilities of the network architecture to handle such changes. One key area to consider is the subscription management process, which governs the user's profile management, authentication, and access control. This process operates centrally, which affects users' security, accessibility, and privacy. Additionally, it increases the system's complexity when handling the large volume of messages sent over networks like IoT. In this work, we propose a Blockchain-based subscription management approach for next generation cellular networks to address the challenges in user profile management and the Authentication and Key Agreement (AKA) process. The method uses a hybrid cryptosystem technique to protect the user's privacy. Based on the evaluation, the system can handle the AKA process with fewer messages passing while improving system availability by utilizing distributed network functions and storage. Finally, we highlight some key points to consider when implementing our proposed approach.**

***Index Terms*—Cellular network, Subscription management, Profile management, AKA procedure, Distributed database, Smart contract, Blockchain.**

I. Introduction

Mobile Network Operators (MNOs) provide high-quality services for the ever-growing users with new demands [1]. In this regard, next-generation networks (i.e., beyond 5G and 6G) are attempting to provide numerous opportunities by integrating various technologies such as Artificial Intelligence (AI), the Internet of Things (IoT), and massive end-to-end connections. Adopting such technologies in current networks raises new concerns about distribution and decentralization, scalability for handling large numbers of users, network complexity, security, user privacy, and performance. [2]–[4]. Among these concerns, the most prominent challenges are ensuring security, availability, and user privacy.

In 5G, the core network consists of different entities depicted in Figure 1, such as resource management, signaling, policy management, subscription management, location management, packet controller, and user plane management. Subscription management involves managing various aspects related to network subscriptions, such as user profiles, registrations, and access control. It consists of three components - Unified Data Registry (UDR), Unified Data Management Function (UDM), and Authentication Server Function (AUSF). As this study focuses on subscriber management, we look at challenges regarding two categories of subscription management:

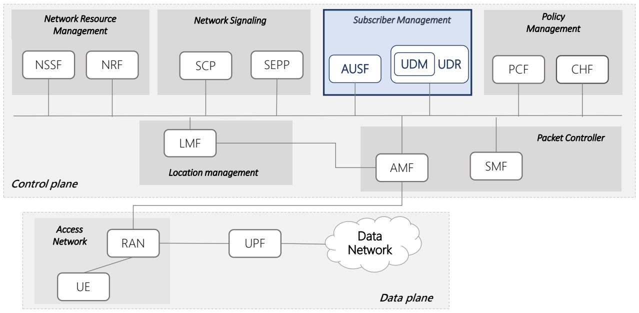

Figure 1. Service-Based Architecture of 5G Cellular Network The focus of this work is on the blue section (subscriber management).

User profile management and data storage: Currently, cellular networks store and access user data in a centralized manner, which raises security and performance concerns. For example, the current UDR can be a single point of failure for user data availability and the target point for user-sensitive data leakage [4]. Furthermore, such a centralized entity can reduce the system's scalability and performance. Also, there may be a violation of user privacy in cloud-based NFVs during storage and transmission [4], and process complexity [4].

Authentication, access control, and key agreement procedure: In addition to data storage and access, the existing authentication and access control procedure handled by AUSF has several issues. The registration step (i.e., the process of authentication, access control, and key agreement) occurs when the user turns on the phone and, it makes the user known to the cellular network. This process can also happen periodically, or during the mobility management procedure. Currently, this method uses the 3GPP-standard Authentication and Key Agreement (AKA) mechanism. Although it can meet the 5G requirement for acceptable latency, it has several challenges that arise for the beyond 5G and 6G cellular networks. For example, in IoT use cases, the inability of a cloud-based centralized authentication mechanism to scale can impact overall system performance [5]. Furthermore, the centralized AKA can be a single point of failure.

In this paper, we propose Blockchain-based subscription management for beyond 5G and 6G cellular networks to deliver trust, reliability, automation of the procedure, and distribution. The main focus of this work is moving the

978-3-903176-59-1 ©2023 IFIP

subscriber management capabilities from the core network to a distributed system supported by Blockchain. Each user in the system will have a specific smart contract that manages the user subscription processes. The user's identity data is stored in a distributed database benefiting from a hybrid cryptosystem to preserve the confidentiality of data and the user's privacy. Although Blockchain can offer several intriguing benefits, it suffers from drawbacks, such as high latency and standardization requirements. So, we also present possible challenges when implementing the approach in the real world, along with several possible improvements.

The **contributions** of the proposed method, can be summarized as follows:

- Providing a secure multi-access to the user data stored in a distributed database, using a hybrid cryptosystem;
- Providing distributed authentication and access control;
- Reducing the number of messages passing in the authentication procedure, resulting in reduced communication and computation overhead;
- Providing forward and backward secrecy of the user's data in the distributed database;
- Real-world implementation without the requirement for additional hardware;
- Reduced load on the network due to migration of subscriber management and authentication functionality to a distributed module.

Paper organization: Section II provides an overview of Blockchain and subscription management in cellular networks. Section III presents related works in authentication and profile management. Section IV explains the motivation for the proposed work, identifying gaps in current approaches. Section V details the proposed method, including its design considerations and technical aspects. Section VI describes the testbed implementation and presents evaluation results. Finally, Section VII discusses the concerns for real-world implementation of the proposed method, potential future directions, and concludes the work.

II. Preliminaries

Authentication and Key Agreement (AKA) procedure provides secure communication between a user device and the mobile network. The user device and the network mutually validate each other's identities during this procedure to establish trust. The device establishes its identification by completing a unique network challenge, indicating that it contains the relevant credentials and secrets. Following successful authentication, both parties generate session keys, which serve as secret codes for encrypting and decrypting the data. These session keys establish the privacy and integrity of the entities and the transferred information.

Blockchain is a peer-to-peer distributed ledger updated only through consensus among the majority of the nodes present on the network [6]. In Blockchain, each block is linked to the previous block by its hash value, and every transaction must be validated by the network's participants using a consensus mechanism [7], [8]. Smart Contracts [9] are a set of rules written as computer codes that are automatically executed, on top of the Blockchain, based on several conditions. Smart Contracts eliminate the need for intermediaries and enable trusted and transparent peer-to-peer transactions.

III. Related Works

In this section, we provide a brief overview of the existing work done in authentication domain for cellular network.

Haddad et. al, [10] proposed a Blockchain-based mutual authentication and key agreement protocol for the 5G network to address the security challenges with existing AKA protocol. Yang et. al [11] present a trusted authentication framework using Blockchain in cloud radio over fiber networks for 5G. Moreover, Goswami et. al, [3] proposed an authentication procedure using Blockchain technology for IoT devices in a 5G network. Another Blockchain-based authentication protocol for 5G networks is proposed by Hojjati et. al [2] who utilize Blockhain as a secure medium for information exchange between home networks and other operators and uses smart contracts to implement access control mechanisms. Chow et. al [12] propose a Blockchain-based authentication and key agreement scheme for 5G networks to address the security issues in existing authentication protocols. Gao et. al [13] propose a Blockchain-based asymmetric authentication and key agreement protocol (BC-AKA) for distributed 5G core networks. Yazdinejad et. al [14] suggest using software-defined networking and Blockchain technology to eliminate the need for repeated authentication during the handover process between heterogeneous cells.

IV. Motivations

We present the existing challenges in subscription management in telecommunication regarding *scalability*, *performance*, *security and privacy*, and *interoperability and integration*.

- **Scalability**: The existing centralized storage and authentication functions have limited capability to scale up regarding the number of users, devices, services, etc. in the telecommunication service. However, beyond 5G/6G networks are expected to handle massive number of devices and connections (e.g., IoT sensors, user devices, and vehicles), leading to a significant increase in the volume of subscriber data.
- **Performance**: The centralized architecture of conventional MNOs results in handling all the connections through a centralized party. This model increases the processing load and overhead in the central point, reducing the quality of service (QoS) and increasing the complexity of IT operations.
- **Security and Privacy**: Protecting the privacy and security of the subscribers (e.g., personal data, credentials, subscribed services, keys, etc.) is crucial. However, existing storage systems are vulnerable to single points of failure, data loss, privacy violations, and unauthenticated access [15]. Moreover, the complexity of authentication, access control, and data integrity in the conventional architecture of MNO are critical security challenges [5].

The user's personal information in cellular networks is an attractive target for advertisement and intelligence agencies, making privacy a significant concern for the users. Currently, the user's privacy can be violated by the storage systems and third-party applications, end-to-end data transmission through several stakeholders, and storing the user's data in a shared environment [5]. Furthermore, the centralized subscription management suffers from single-point failures, which can affect the availability and fault tolerance of the system [4].

- **Interoperability and integration**: Beyond 5G/6G cellular networks need a high level of collaboration between different components and entities, as well as interoperability between many technologies and systems. Ensuring secure interoperability and collaboration between different entities is vital to providing seamless connections. The existing centralized storage is not able to provide the required interoperability and integration.

Addressing these challenges will require architectural enhancements, new protocols, and the adoption of advanced technologies within the existing UDR, HSS, and AKA procedures in the next-generation networks.

V. Proposed method

This section provides a detailed description of the proposed Blockchain-based subscription management procedure for beyond 5G/6G cellular networks. We introduce all the smart contracts used in the proposed method along with their functionalities. Note that, in the proposed method, we assume that 1) all off-chain connections (i.e., outside of the Blockchain) are secure, and 2) User equipment supports a novel SIM card designed for beyond 5G networks in which the user Blockchain address ($Addr_u$) and public/ private key pair (Pub_u, Pr_u) are hard-coded.

A. Designed smart contracts

Here, we provide the description and data model of the designed smart contracts to handle subscription management.

1) User List contract (SC_{UL}): This smart contract stores the list of users who are registered in the system. The user list is stored in a mapping of the user's address ($Addr_u$) to its subscription status (i.e., not subscribed, subscription in process, and subscribed).

2) Subscription contract (SC_{Sub}): The subscription smart contract is dedicated to handling the user subscription procedure in the host MNO. After receiving and validating the user's Subscription request (similar to the user's SIM-card activation procedure in the current cellular network), this contract activates/deploys the user's unique smart contract (SC_U) and updates her status in SC_{UL}. Note that the details of the subscription procedure will be described in the next subsections. Moreover, the other important function of this smart contract is to delegate/revoke the ownership of the user's contract to the host MNO for updating her data in IPFS or SC_U (e.g., when the user switches between MNOs, the recipient MNO needs to have the ownership of `update()` function in SC_U, while the ownership of donor MNO needs to be revoked).

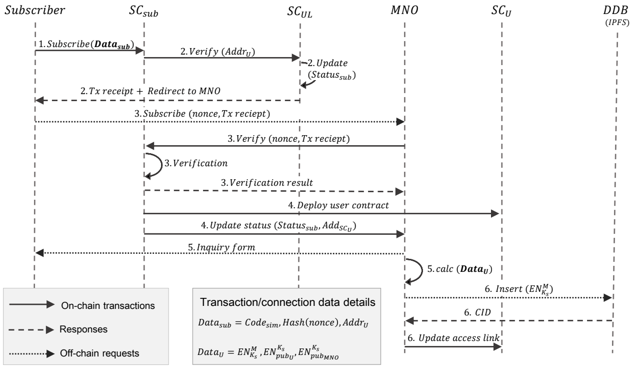
Subscriber
SC_{sub}
SC_{UL}
MNO
SC_U
DDB (IPFS)
1. Subscribe($\boldsymbol{Data_{sub}}$)
2. Verify ($Addr_U$)
2. Update ($Status_{sub}$)
2. Tx receipt + Redirect to MNO
3. Subscribe (nonce, Tx reciept)
3. Verify (nonce, Tx reciept)
3. Verification
3. Verification result
4. Deploy user contract
4. Update status ($Status_{sub}, Add_{SC_U}$)
5. Inquiry form
5. calc ($\boldsymbol{Data_U}$)
6. Insert ($EN^M_{K_s}$)
6. CID
6. Update access link
On-chain transactions
Responses
Off-chain requests
Transaction/connection data details
$Data_{sub} = Code_{sim}, Hash(nonce), Addr_U$
$Data_U = EN^M_{K_s}, EN^{K_s}_{pub_U}, EN^{K_s}_{pub_{MNO}}$

Figure 2. User subscription procedure

3) User contract (SC_U): A user smart contract is a unique smart contract deployed for a particular user that stores, at least, a mapping of the user's address ($Addr_u$) to their current balance in Blockchain wallet, $CID_{EN^M_{K_s}}$ that is the access identifier of IPFS storage to the user's data, $EN^M_{K_s}$ that is is the user's data (M) encrypted by K_s, $Hash(M)$ is the hash of content M, $EN^{K_s}_{Pub_u}$ and $EN^{K_s}_{Pub_{MNO}}$ that are the K_s encrypted by the user's and MNO's public key, respectively; and the user's phone number ($Number_u$).

Note that, since in Blockchain and smart contracts the data is transparent for everyone in the network, none of the user's PII data is stored in SC_U. These data are proposed to be stored confidentially in IPFS.

4) Registration contract (SC_{Reg}): This smart contract manages the user registration procedure to make the user known and traceable for the network. The registration procedure is done once the user turns her phone on, and also on a periodical basis.

5) Authentication and Access Control smart contract (SC_{AAC}): AAC smart contract is designed to manage the user's authentication and access control procedure by validating her eligibility to access the network. In this regard, the smart contract checks if the user is the one who claims to be, the expiration time of the user's contract with MNO, the location of the user to give service and other policies that can be defined by MNO.

B. User subscription and Profile management

The subscription steps of the user (U) are as follows (See Fig. 2).

1) U sends a subscription request to SC_{Sub} by creating a transaction as: $< Code_{sim}, Hash(nonce), Addr_U >$ where $nonce$ is a random number generated by user, and $Hash(nonce)$ is the hash of $nonce$ calculated by $Keccak256$ [16] algorithm. $Code_{sim}$ is a secret code given to the user once she bought the SIM-Card, and $Addr_U$ is the user's Blockchain address hardcoded in the SIM-Card.

2) Once SC_{Sub} receives the request, stores $Hash(nonce)$ and asks SC_{UL} to verify the subscription status of $Addr_U$. If the user is not subscribed, SC_{UL} updates the user's $Status_{sub}$ to 1 which means `'verified for activation'`. Finally, SC_{UL} sends the transaction receipt to SC_{Sub} to send it to the user.
3) Once U is redirected to the subscription page of the MNO, she sends $< Tx - reciept, nonce >$. MNO can verify the request from SC_{Sub} by calling `Verify()` function of SC_{Sub} and sending $< Tx - reciept, nonce >$ to its arguments. If the following conditions pass, the `Verify()` function would return `true` as the indication of successful verification: $Hash^T(nonce) == Hash(nonce)$; $Status_{sub} == 1$ $Hash^T(nonce)$ is the hash of $nonce$ received by MNO, and $Hash(nonce)$ has been stored in SC_{Sub} in *Step 2*.
4) SC_{Sub} deploys a unique smart contract for the user (SC_U) and changes $Status_{sub}$ to `'subscribed'`. Then, create an event for MNO to confirm the user's subscription by sending $< Status_{sub}, Addr_{SC_U} >$. Moreover, it inserts $Addr_{SC_U}$ in SC_{UL}.
5) Once receiving the confirmation, MNO sends the subscription form to the user and receives the user's identity data (M). Because the data will be stored in a distributed database (i.e., IPFS), after receiving M, MNO needs to strictly limit the access to data. **Note that, using IPFS in the procedure is to address the scalability and storage requirements of Blockchain.** The only external entities that can have access to data are the U and MNO. To do so, we employed a hybrid cryptosystem for a multi-user environment. The hybrid cryptosystem is a technique of combining symmetric and asymmetric cryptography algorithms (e.g., PGP, Pretty good privacy, algorithm). To apply this method, MNO executes the following steps:
   - Generates symmetric key K_s;
   - Encrypts K_s using Pub_U and Pub_{MNO} and gets $EN^{K_s}_{Pub_U}$ and $EN^{K_s}_{Pub_{MNO}}$
   - Encrypts M with K_s to get $EN^M_{K_s}$
6) MNO needs to modify $Attr_U$. So, it requests SC_{Sub} to execute the write delegation procedure by sending the $Addr_U$ to it. SC_{Sub} retrieves the user request and verifies the following condition: $Status_{sub} == 2$. If the validation is successful, MNO will be able to update data.
7) MNO stores $EN^M_{K_s}$ in IFPS as a distributed database. After storing the data in IPFS, it would be indexed by a cryptographic hash function, which results in returning its unique content identifier (CID) to MNO. The CID (let's call it $CID_{EN^M_{K_s}}$) can be used for further access to the data in IPFS. Moreover, MNO can store $Attr_U$ containing $EN^{K_s}_{Pub_U}$ and $EN^{K_s}_{Pub_{MNO}}$ into SC_U. For further user connections, MNO can verify the user's address, and get its profile from IPFS. Using this procedure, the user profile can be retrieved either by the user or the MNO. The other entities won't be able to have access to plain-text user data.

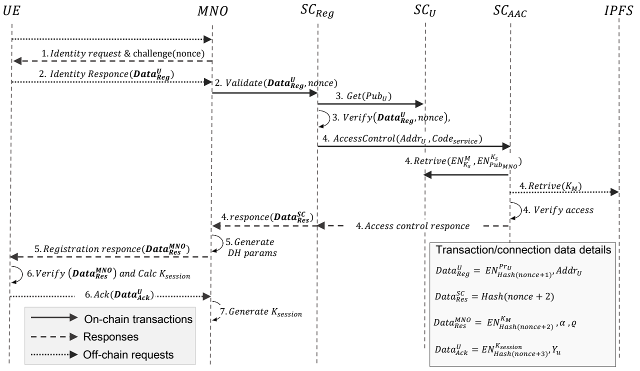
UE
MNO
SC_{Reg}
SC_U
SC_{AAC}
$IPFS$
1. Identity request & challenge(nonce)
2. Identity Responce($\mathbf{Data}^U_{Reg}$)
2. Validate($\mathbf{Data}^U_{Reg}$, nonce)
3. Get(Pub_U)
3. Verify($\mathbf{Data}^U_{Reg}$, nonce),
4. AccessControl($Addr_U$, $Code_{service}$)
4. Retrive($EN^M_{K_s}$, $EN^{K_s}_{Pub_{MNO}}$)
4. Retrive(K_M)
4. Verify access
4. responce($\mathbf{Data}^{SC}_{Res}$)
4. Access control responce
5. Generate DH params
5. Registration responce($\mathbf{Data}^{MNO}_{Res}$)
6. Verify ($\mathbf{Data}^{MNO}_{Res}$) and Calc $K_{session}$
6. Ack($\mathbf{Data}^U_{Ack}$)
7. Generate $K_{session}$
On-chain transactions
Responses
Off-chain requests
Transaction/connection data details
$Data^U_{Reg} = EN^{Pr_U}_{Hash(nonce+1)}, Addr_U$
$Data^{SC}_{Res} = Hash(nonce + 2)$
$Data^{MNO}_{Res} = EN^{K_M}_{Hash(nonce+2)}, \alpha, \varrho$
$Data^U_{Ack} = EN^{K_{session}}_{Hash(nonce+3)}, Y_u$

Figure 3. User registration (authentication) and Key-agreement procedure

C. Blockchain-based Authentication and Key-management

The user registration procedure introduces the user to the network to make it capable of finding the user. The initial registration would be executed when the users turn on their phone. Periodic registrations are required to keep the user known for the network. Here, we describe Blockchain-based registration along with the session key agreement procedure. Note that, to decrease the complexity and latency, both registration and key-agreement processes are done in one phase. The procedure is explained in Figure 3.

1) Firstly, U sends a registration request to MNO. After receiving the user's request, MNO responds by asking to send the identification data and encrypting a challenge (i.e., $nonce$) with Pr_U.
2) U calculates $Hash(nonce + 1)$ using Keccak-256 and signs the result with her private key. Finally, she responds MNO by encapsulating the encrypted hash along with $Addr_U$ (in plain text). The response is as follows:

$$Data^U_{reg} = [EN^{Pr_{UE}}_{Hash(nonce+1)}, Addr_U]$$

After receiving the response, MNO calls `Validate()` function of SC_{Reg} by transmitting $Data^U_{reg}$ and $nonce$ (let's call it $nonce^t$), to verify the request.
3) SC_{Reg} gets the address of SC_U from SC_{UL} and retrieves Pub_U. Then the verification procedure is as follows:

$$Hash(nonce^t + 1) == DE^{Pub_{UE}}_{EN^{PR_{UE}}_{Hash(nonce+1)}}$$

4) If validation was successful, SC_{Reg} requests SC_{AAC} to verify the user's access permissions by sending $< Addr_U >$. Since MNO has access to the user profile, SC_{AAC} can retrieve $EN^{K_s}_{Pub_{MNO}}$ from SC_U through MNO, to find the symmetric key to decrypt the user data. Then it retrieves $EN^M_{K_s}$ from IPFS and computes $M = DE^{K_s}_{EN^M_{K_s}}$. Using the user's data, SC_{AAC} verifies if the user's subscription has not expired, the user is

eligible to have access from the specific geographical location, and the user's balance is sufficient.

Once SC_{Reg} gets the access control result, it responds to MNO by sending $Hash(nonce+2)$. If the validation was not successful, SC_{Reg} would send the deny response.

5) To execute the mutual authentication and create a session key for further connections, MNO selects three parameters of Diffie-Hellman [17] key agreement algorithm namely, α, ϱ, and a private key X_{MNO}. Using these parameters, MNO calculates its session public key,Y_{MNO} and transmits the following response to the user:

$$< \alpha, \varrho, Y_{MNO}, EN^{K_M}_{Hash(nonce+2)} >$$

6) Once receiving the reply, U verifies:

$$Hash(nonce+2) == DE^{K_M}_{EN^{K_M}_{Hash(nonce+2)}}$$

Since K_M is known for only the U and MNO, if $Hash(nonce+2)$ was valid, we can claim that the sender is trusted. Then, the U chooses a Diffie-Hellman private key,X_u, and using α and ϱ calculates its public key (Y_u). Next, U calculates the session key, K_s, with the use of Y_{MNO} and X_u. Finally, the user sends an acknowledgment to MNO concerning successful verification and accepting the connection. To do so, it encrypts $Hash(nonce+3)$ with K_s and encapsulates the result with Y_u in plain text.

7) Once receiving the acknowledgment from U in MNO, it recalculates the session key (let's call it K'_s) using Y_u, and X_{MNO}. Then, using the generated key, MNO can validate the following equation for final authentication:

$$Hash(nonce+3) == DE^{K'_s}_{EN^{K_s}_{Hash(nonce+3)}}$$

Note that, up until the last step, nothing is written in the Blockchain. So, it is not affected by the Blockchain's consensus latency.

VI. Evaluation

In this section, we evaluate the performance of the proposed system (user subscription and authentication and key management procedure). First, we describe in detail the implementation of the testbed, followed by the evaluation of the system performance in regard to the following indicators:

- *Comparison with existing solutions:* Here, we evaluated the number of messages passing in our proposed system (i.e., Network overhead), security features (e.g., resistance against common attacks, privacy issues, etc.), and implementation scenario compared with the existing AKA procedure in 4G and 5G, and several proposed methods in the state of the arts.
- *Smart contract cost:* This evaluation is related to the GAS cost for executing the transactions in Blockchain and smart contracts.
- *Computational overhead:* This evaluation considers the latency of writing and reading operations in Blockchain and the distributed database.

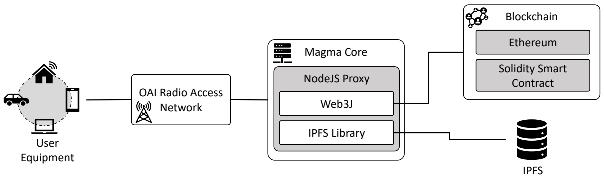

Figure 4. An architectural overview of our proposed scheme.

A. Testbed implementation

To evaluate the performance of the proposed method, we designed a use case scenario in which, first the user will subscribe to the MNO by sending their identity data. Then this identity data will be stored in a distributed database in a secure format, and only the user and MNO can have access to this data. Then, we perform the user registration (i.e., authentication, access control, and key agreement) procedure.

For the evaluation, we implemented the testbed with the architecture depicted in Fig. 4. The key modules of this prototype consist of request handling, subscription, authentication, access control, key agreement, and Read/write module.

B. Comparison with existing solutions

In this section, we compare the proposed method with the existing solutions. The summary of the comparison is listed in Table I and explained as follows:

- *Message number:* This indicator compares the number of messages passing between MNO and user to establish a connection in the registration (i.e., authentication and access control) procedure. Indeed, the lower number of messages passing in the registration procedure results in lower latency and network overhead. Note that, since the subscription procedure needs to be done only for one time, the number of messages passing in this case does not have a significant effect on the network overhead. So, our assessment is mostly focused on the authentication and key agreement procedures.
- *Implementation model (Architecture type):* Indicates the architecture of subscription management procedure that can be centralized (C) or decentralized (D).
- *Privacy-preserving:* This feature indicates if the proposed method preserves the privacy and confidentiality of the user's data in the network.
- *Storage type:* Indicates the type of storage used for recording the user data, it can be centralized or distributed.
- *The number of write operations to Blockchain:* This is the number of transactions sent to the Blockchain to update the state of Blockchain. These transactions need to be validated through consensus procedure and they have the validation overhead in the network. In this scenario, the

Table I
COMPARISON

Param. Ref.	Message #	Model	Privacy	Storage	Write #	Read #
[3]	4	C	No+	D	1	4
[2]	4	D	No	C	2	-
[10]	6	D	Same as 5G	C	6	6
[12]	5	D	Same as 5G	C	1	1
[13]	5	C*	Same as 5G	C	2	3
This work	**4**	**D**	**Yes****	**D**	**1**	**3**

Note: In “model” and “storage” columns, ‘C’ represents “Centralized”, and ’D’ represents “Decentralized” .
+ All user-identifiable data are stored in Blockchain
* keys are stored in Blockchain
** Hybrid crypto-system is used to confidentially store the user’s data in IPFS. Moreover, while transmitting data, all are encrypted with asymmetric keys. The user also has read access to her identity data in IPFS.

Table II
TIME CONSUMED BY EACH ACTION

Process	Functions	Description	Symbol	Time (ms)
On-chain	Validation	New user validation	T_{valid}	108.57
	Add	Adding user Updating data	T_{add} T_{upIPFS}	263.35 106.19
	MNO claim	Adding the request	T_{claim}	275.32
	AAC	Authentication and access control	T_{AAC}	119.28
	Update	Updating data	T_{update}	272.12
Off-chain	Encryption	Symmetric encryption Asymmetric encryption Symmetric decryption Asymmetric decryption	T_{SEn} T_{AsEn} T_{SDe} T_{AsDe}	0.50 0.26 0.04 0.04
	Hashing	Hash calculation	T_{hash}	0.77
	IPFS	Read Write	T_{read} T_{write}	3.39 30.55

lower number of transactions, mostly, in the registration procedure results in lower latency and network overhead.

- *The number of read operations from Blockchain:* This is the number of transactions sent to the Blockchain to read a value from Blockchain. These transactions do not have any time overhead on the network, since they don’t update any state of the Blockchain.

C. Computational overhead evaluation

We evaluate the computational overhead of the proposed subscription management solution in terms of its latency. The latency of the method is assessed for 1) the on-chain functions, 2) the network and computation latency (i.e., off-chain processes), and 3) theoretical estimation for the system latency in real-world implementation using private cellular network [18].

On-chain and off-chain process latency: Table II lists the latency of the function calls and executions for both on-chain and off-chain processes. Note that, the **latency of off-chain processes** is evaluated in a system with a Core i7 CPU, 16GB RAM, and 128GB hard disk, and for the **on-chain process**, we send 50 requests **simultaneously** to the Blockchain and calculated the **average delay** of all transactions after receiving the transaction receipt for all of the requests. As shown in the table, the latency of the on-chain procedure is significantly higher than the off-chain procedure. Indeed, this latency is expected, since the consensus convergence is needed to update the ledger state. The discussion on the latency of the Blockchain, and possible solutions to overcome it is provided in Section VII. However, in the proposed method we minimized the writing operation in the Blockchain.

Theoretical estimation for the system latency: Since the subscription procedure needs to be done only at the time of user subscription in the MNO, its latency does not have any effect on the network performance. So, in this section, we theoretically calculate the expected latency for the proposed system (including all required on-chain and off-chain executions). Based on Fig. 3, the user and MNO will experience the T_{user}, T_{MNO}, respectively, as follows:

$$T_{user} = 3T_{AsEn} + 3T_{AsDe} + 5T_{hash} + T_{SEn} + T_{Claim} + T_{AAC}$$

$$T_{MNO} = T_{user} + T_{SDe} + T_{hash} + T_{write} + T_{update} + T_{SEn}$$

It means the user’s experienced latency is $\simeq 400ms$, and for the MNO this amount is $\simeq 700ms$. Indeed this latency is significantly higher than the expected latency in cellular networks. The solutions to improve the system are discussed in Section VII.

VII. DISCUSSION AND FUTURE DIRECTION

This paper presents a Blockchain-based subscription management and authentication (registration) system for beyond 5G networks. The proposed solution can minimize IT complexity, offer high security, and eliminate centrally managed entities. However, we must consider the following factors while implementing this method in the real world.

- **Owner of the Blockchain:** The proposed method is limited to a single MNO, which aims to utilize the distributed solution for the subscription management process. Thus, the MNO serves as the owner of the Blockchain. However, a consortium of the MNO and all other service providers collaborating with the MNO (for e.g., MVNOs that work with the MNO) can also acquire the ownership.
- **Participants for securing the Blockchain:** In the proposed method, users can only read the Blockchain and their specific smart contract. Due to their restricted processing, storage, and resource capabilities, the participation of the network user in the consensus process is not only ineffective, but it also runs the risk of creating new attack vectors and security breaches.
- **Latency and storage:** The Blockchain has a trilemma of Scalability, Security, and Decentralization, which means,

this technology cannot provide all these features together [19]. In our scenario, the most prominent challenge is the system's scalability in terms of latency and storage. We can address these challenges as follows:

  - Researchers in the Blockchain community may consider *designing a Blockchain for cellular networks* with particular consensus models, block sizes, transaction fees, block times, incentives, and other specifications to enable verifying a greater volume of transactions in a given time.
  - *Chain Sharding* is another option where a chain is split into smaller partitioned chains to divide the transaction loads [20]. Even with existing consensus models (PoS and PBFT), this technique boosts performance while reducing storage usage. Several sharding solutions are demonstrating the feasibility of this approach. For instance, compared to $15 - 20$ TpS (Transactions per Second) in Ethereum, RapidChain [21] raised throughput to 7380 TpS. According to [22], sharding is a promising method for improving scalability to a level suitable for cellular network requirements.

- **Data Privacy:** When data, identity, or location are compromised, it can cause user privacy violations [5]. The *preservation of metadata privacy* [23] is one of Blockchain's fundamental features. It involves using a random address to hide the actual identity of the senders and receivers of transactions (in our use case, users and providers). Our proposed method addresses the user privacy requirements in the following ways: 1) none of the user's identifiable data, such as $SUPI$ in current 5G networks, is transmitted or stored in clear text; 2) the user information is stored in IPFS using a hybrid cryptosystem method that allows only the user and MNO to access these data; 3) Since we implement Blockchain in the core network, outside intruders do not have access to any internal Blockchain transactions. They can only actively/passively eavesdrop on the message passing (which is protected by several techniques). 4) To ensure content privacy, we only preserved the minimal non-PII data required to deliver connectivity and payment to the user (the rest is in IPFS).

References

[1] "Global mobile trends 2021- navigating covid-19 and beyond," Global System for Mobile Communications Association, Tech. Rep., 12 2020. [Online]. Available: https://data.gsmaintelligence.com/api-web/v2/research-file-download?file=141220-Global-Mobile-Trends.pdfid=58621970

[2] M. Hojjati, A. Shafieinejad, and H. Yanikomeroglu, "A blockchain-based authentication and key agreement (aka) protocol for 5g networks," *IEEE Access*, vol. 8, pp. 216 461–216 476, 2020.

[3] B. Goswami and H. Choudhury, "A blockchain-based authentication scheme for 5g-enabled iot," *Journal of Network and Systems Management*, vol. 30, no. 4, p. 61, 2022.

[4] D. C. Nguyen, P. N. Pathirana, M. Ding, and A. Seneviratne, "Blockchain for 5g and beyond networks: A state of the art survey," *Journal of Network and Computer Applications*, vol. 166, p. 102693, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804520301673

[5] R. Khan, P. Kumar, D. N. K. Jayakody, and M. Liyanage, "A survey on security and privacy of 5g technologies: Potential solutions, recent advancements, and future directions," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 1, pp. 196–248, 2019.

[6] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[7] S. Bouraga, "A taxonomy of blockchain consensus protocols: A survey and classification framework," *Expert Systems with Applications*, vol. 168, p. 114384, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417420310587

[8] W. Wang, D. T. Hoang, P. Hu, Z. Xiong, D. Niyato, P. Wang, Y. Wen, and D. I. Kim, "A survey on consensus mechanisms and mining strategy management in blockchain networks," *IEEE Access*, vol. 7, pp. 22 328–22 370, 2019.

[9] V. Y. Kemmoe, W. Stone, J. Kim, D. Kim, and J. Son, "Recent advances in smart contracts: A technical overview and state of the art," *IEEE Access*, vol. 8, pp. 117 782–117 801, 2020.

[10] Z. Haddad, M. M. Fouda, M. Mahmoud, and M. Abdallah, "Blockchain-based authentication for 5g networks," in *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*. IEEE, 2020, pp. 189–194.

[11] H. Yang, H. Zheng, J. Zhang, Y. Wu, Y. Lee, and Y. Ji, "Blockchain-based trusted authentication in cloud radio over fiber network for 5g," in *2017 16th international conference on optical communications and networks (ICOCN)*. IEEE, 2017, pp. 1–3.

[12] M. C. Chow and M. Ma, "A secure blockchain-based authentication and key agreement scheme for 3gpp 5g networks," *Sensors*, vol. 22, no. 12, p. 4525, 2022.

[13] Z. Gao, D. Zhang, J. Zhang, Z. Liu, H. Liu, and M. Zhao, "Bc-aka: Blockchain based asymmetric authentication and key agreement protocol for distributed 5g core network," *China Communications*, vol. 19, no. 6, pp. 66–76, 2022.

[14] A. Yazdinejad, R. M. Parizi, A. Dehghantanha, and K.-K. R. Choo, "Blockchain-enabled authentication handover with efficient privacy protection in sdn-based 5g networks," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 2, pp. 1120–1132, 2019.

[15] M. Tahir, M. H. Habaebi, M. Dabbagh, A. Mughees, A. Ahad, and K. I. Ahmed, "A review on application of blockchain in 5g and beyond networks: Taxonomy, field-trials, challenges and opportunities," *IEEE Access*, vol. 8, pp. 115 876–115 904, 2020.

[16] G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche, "The keccak sha-3 submission," STMicroelectronics, 2NXP Semiconductors, Tech. Rep., 2011. [Online]. Available: https://keccak.team/index.html

[17] E. Rescorla *et al.*, "Diffie-hellman key agreement method," 1999. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc2631

[18] N. Aryal, F. Ghaffari, S. Rezaei, E. Bertin, and N. Crespi, "Private cellular network deployment: Comparison of openairinterface with magma core," in *2022 18th International Conference on Network and Service Management (CNSM)*, 2022, pp. 364–366.

[19] A. Hafid, A. S. Hafid, and M. Samih, "Scaling blockchains: A comprehensive survey," *IEEE Access*, vol. 8, pp. 125 244–125 262, 2020.

[20] G. Kaur and C. Gandhi, "Scalability in blockchain: Challenges and solutions," in *Handbook of Research on Blockchain Technology*. Elsevier, 2020, pp. 373–406.

[21] M. Zamani, M. Movahedi, and M. Raykova, "Rapidchain: Scaling blockchain via full sharding," in *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, 2018, pp. 931–948.

[22] G. Wang, Z. J. Shi, M. Nixon, and S. Han, "Sok: Sharding on blockchain," in *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*, 2019, pp. 41–61.

[23] K. Yue, Y. Zhang, Y. Chen, Y. Li, L. Zhao, C. Rong, and L. Chen, "A survey of decentralizing applications via blockchain: The 5g and beyond perspective," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2191–2217, 2021.