sensors

MDPI

Article

A Blockchain-Assisted Security Protocol for Group Handover of MTC Devices in 5G Wireless Networks

Ronghao Ma [1], Jianhong Zhou [1] and Maode Ma [2,*]

[1] School of Computer and Software Engineering, Xihua University, Chengdu 610039, China; 212022085400073@stu.xhu.edu.cn (R.M.); zhoujh@uestc.edu.cn (J.Z.)
[2] KINDI Center for Computing Research, College of Engineering, Qatar University, Doha P.O. Box 2713, Qatar
* Correspondence: acadmmd@gmail.com

Abstract: In the realm of the fifth-generation (5G) wireless cellular networks, renowned for their dense connectivity, there lies a substantial facilitation of a myriad of Internet of Things (IoT) applications, which can be supported by the massive machine-type communication (MTC) technique, a fundamental communication framework. In some scenarios, a large number of machine-type communication devices (MTCD) may simultaneously enter the communication coverage of a target base station. However, the current handover mechanism specified by the 3rd Generation Partnership Project (3GPP) Release 16 incurs high signaling overhead within the access and core networks, which may have negative impacts on network efficiency. Additionally, other existing solutions are vulnerable to malicious attacks such as Denial of Service (DoS), Distributed Denial of Service (DDoS) attacks, and the failure of Key Forward Secrecy (KFS). To address this challenge, this paper proposes an efficient and secure handover authentication protocol for a group of MTCDs supported by blockchain technology. This protocol leverages the decentralized nature of blockchain technology and combines it with certificateless aggregate signatures to mutually authenticate the identity of a base station and a group of MTCDs. This approach can reduce signaling overhead and avoid key escrow while significantly lowering the risk associated with single points of failure. Additionally, the protocol protects device anonymity by encrypting device identities with temporary anonymous identity markers with the Elliptic Curve Diffie–Hellman (ECDH) to abandon serial numbers to prevent linkage attacks. The resilience of the proposed protocol against predominant malicious attacks has been rigorously validated through the application of the BAN logic and Scyther tool, underscoring its robust security attributes. Furthermore, compared to the existing solutions, the proposed protocol significantly reduces the authentication cost for a group of MTCDs during handover, while ensuring security, demonstrating commendable efficiency.

Keywords: the fifth-generation cellular network; group handover authentication; MTCD; blockchain

Citation: Ma, R.; Zhou, J.; Ma, M. A Blockchain-Assisted Security Protocol for Group Handover of MTC Devices in 5G Wireless Networks. *Sensors* **2024**, *24*, 2331. https://doi.org/10.3390/s24072331

Academic Editor: Zahir M. Hussain

Received: 19 February 2024
Revised: 30 March 2024
Accepted: 1 April 2024
Published: 6 April 2024

1. Introduction

Machine-type communication (MTC), also known as machine-to-machine (M2M) communication, has been obtaining increasing attention for the widespread adoption of the fifth generation (5G) wireless cellular networks. With potential to revolutionize a broad spectrum of sectors like healthcare, logistics, manufacturing, process automation, energy, and utilities, MTC stands at the forefront of technological advancement. In certain communication scenarios, such as high-speed trains, convoys, and buses, multiple MTC devices (MTCDs) may move from the coverage area of one base station to another. This occurrence is labeled as a "group handover", necessitating each MTCD to authenticate during the transition [1].

The 5G wireless network employs a multitude of miniature millimeter-wave cellular base stations, aiming at serving a large number of users. This approach allows for the efficient reuse of limited spectrum resources and supports the access of large-scale MTC

devices. Furthermore, adding more base stations can effectively alleviate traffic congestion in wireless channels. Therefore, the 5G wireless network is expected to significantly enhance the performance of wireless connectivity with higher transmission rates, lower communication latency, and greater network capacity. Considering the impending 5G wireless network evolution, a prominent challenge emerges concerning access authentication and data transmission for a vast array of IoT terminals [2]. If each IoT device persists in employing the Extensible Authentication Protocol-Authentication and Key Agreement (EAP-AKA) or the 5G Authentication and Key Agreement (5G-AKA) for its authentication, it would inevitably lead to a surge in signaling and communication overheads [3]. The 5G-AKA protocol is a new authentication and key agreement protocol standardized by the 3rd Generation Partnership Project (3GPP) to be used in the 5G wireless networks. The 5G-AKA protocol plays a pivotal role in enhancing 5G wireless network security by enabling mutual authentication between base stations, core networks, and user devices, thus overcoming the vulnerabilities identified in the 4G wireless networks. It introduces the Subscription Concealed Identifier (SUCI), leveraging encryption to safeguard the sensitive information of users like International Mobile Subscriber Identification (IMSI), against unauthorized interception and malicious base station attacks. This approach significantly bolsters personal privacy and network security, ensuring wireless communications are secure and confidential. Notably, the existing 5G standard, as outlined in the 3GPP standard, exhibits challenges when dealing with concurrent handovers of a cluster of MTCDs. Recent studies [4] have highlighted security loopholes within the handover authentication procedure. These encompass the absence of reciprocal authentication, deficiencies in Key Forwarding Security (KFS), and a heightened vulnerability to Denial of Service (DoS) attacks. Moreover, due to the ultra-dense nature of 5G networks with a larger number of cells, the handover events occur more frequently, leading to increased signaling when a group of MTCDs simultaneously handover from a service base station to a target base station.

Current research has not delivered an efficient and secure solution for MTCD group authentication in 5G wireless networks. Blockchain technology offers decentralization to enhance data security and privacy while shifting authentication from centralized servers to base stations, effectively countering both DoS and Distributed Denial of Service (DDoS) attacks. To address the abovementioned shortcomings, we designed a blockchain-assisted security protocol for group handover (BSPGH) for MTCDs in 5G wireless networks. By uniquely combining blockchain technology with group authentication in 5G wireless networks, the protocol ensures secure handover authentication while remaining simple to deploy with high efficiency. The specific contributions made in this paper can be summarized as follows:

1. We have devised the BSPGH protocol, leveraging the capabilities of blockchain technology. This protocol guarantees the preservation of all security attributes while remaining in alignment with the architecture of the 5G wireless network specified by the 3GPP standard and ensuring its suitability for the MTCD handover scenarios.
2. The proposed BSPGH protocol harnesses the blockchain to establish a decentralized public key management system. It directly culminates in the realization of mutual authentication between base stations and MTCDs. It adeptly streamlines the handover authentication procedure, curtails the volume of interaction messages, safeguards against single points of failure, and fortifies resistance against both DoS and DDoS attacks.
3. The BSPGH solution is built upon the Reliable Malicious KGC-Resistant Certificateless Aggregate Signature (RelCLAS) algorithm and undergoes a formal assessment using the Burrows-Abadi-Needham (BAN) logic and Scyther Tool. It can achieve mutual authentication with key negotiation, anonymity, traceability, perfect forward and backward secrecy, resilience against DoS attacks, and defense against impersonation attacks, etc.

The structure of the remaining paper is as follows. Section 2 provides an overview of the existing handover authentication schemes. Section 3 explains the background

knowledge. Section 4 explains the system model and attack model. Section 5 describes the details of the proposed protocol. Section 6 presents the security analysis of the protocol. Section 7 evaluates the performance of the solution. Finally, Section 8 concludes the paper with a summary.

2. Related Work

To address the challenges posed by the technologies in 5G wireless networks, the authors in [5] employed aggregated message authentication codes (AMAC) to reduce signaling overheads. By this approach, the group leader aggregates message authentication codes (MACs) from all group members and sends the aggregated information to the network. However, the protocol fails to ensure user privacy because the messages are transmitted in plain text over insecure channels. The solutions in [6,7] need bilinear mapping calculations, which can result in higher computational costs. In [6], a lightweight and efficient group authentication protocol is proposed. This scheme integrates bilinear mapping and aggregate certificateless signature mechanisms to address the real-time secure and efficient access of multiple MTCDs. In [7], a multi-user access authentication scheme has been proposed, which leverages the features of a network architecture combining Mobile Edge Computing (MEC) and Software-Defined Networking (SDN) to perform pre-authentication by predicting a potential target base station for handover. However, this system architecture introduces deployment challenges in practice, and it involves modular exponentiations, which need higher computational costs. In [8], a lightweight group identity authentication scheme is proposed, suitable for both centralized and decentralized settings. It enables all MTCDs to negotiate and generate a group key as a session key for mutual communication. However, this scheme is susceptible to DoS attacks and has high energy consumption when using bilinear pairings. In [1], a privacy-preserving handover authentication protocol suitable for a group of MTCDs in 5G networks is proposed. The protocol aims to reduce signaling costs by aggregating messages from two MTCDs with an aggregated MAC and sending them through an authenticated group member. However, this scheme is susceptible to DDoS attacks.

The solutions in [9–13] all employ blockchain-based techniques for identity authentication. Among them, ref. [9] introduces a blockchain-based protocol to achieve mutual authentication and session key negotiation for vehicles. It is accomplished by introducing an auxiliary blockchain and a parent blockchain, along with the use of an Interplanetary File System (IPFS) to collaborate in information storage. The authentication process by this scheme incorporates elliptic curve cryptography (ECC) and one-way hash functions. The introduction of multiple blockchains may lead to unnecessary resource consumption. Ref. [10] presents a group-based handover authentication scheme for 6G heterogeneous networks, leveraging blockchain for storing authentication information and utilizing aggregate signatures for streamlined batch user authentication. Ref. [11] introduces a collaborative authentication scheme using blockchain in heterogeneous networks. This scheme improves the SM9 algorithm and proposes a verifiable user identity legitimacy through group signature, eliminating data redundancy caused by unfiltered blockchain information in wireless communication. Ref. [12] proposes a lightweight blockchain-based initial and handover authentication protocol for vehicles and infrastructure. This protocol involves vehicles performing lightweight calculations using hash and XOR operations, and the information required for handover authentication is stored in Roadside Units (RSUs) through secure sharing within the consortium blockchain. It can revoke unauthorized vehicles directly using blockchain without the need for a third-party entity. Ref. [13] presents a blockchain-based group key distribution method, distributing and updating group session keys among group members using smart contracts for identity authentication. The authors in [14] have proposed two protocols tailored to different security requirements. However, these protocols are only applicable to scenarios with a predefined trajectory. Ref. [15] proposes a pre-handover authentication mechanism based on the Chinese Remainder Theorem (CRT), allowing user terminals to achieve rapid handover authentication and key negotiation with

the target access relay node. However, it is only applicable to fixed trajectory communication in high-speed rail contexts. Ref. [16] produces a group MTCD handover authentication using base stations installed on drones. It applies to extremely specific scenarios and is not suitable for general use. At the same time, there is also the issue of drones used as base stations being unable to sustain long-term energy consumption. The solutions in [5,6,8,10,13,16] are all susceptible to the risk of DoS attacks. During the aggregation of information, the aggregated information can only be successfully verified if all members are legitimate. Attackers can send false aggregate information and intentionally cause the entire group's verification to fail. Ref. [17] presents a secure and privacy-preserving handover scheme for 5G networks, but it comes with higher computational costs due to the use of multiple modular exponentiation algorithms.

The development of 5G networks has facilitated the support for large-scale connections of MTC devices, offering higher data rates, lower latency, greater connection capacity, and higher energy efficiency. The introduction of large-scale MTC devices enables tens of thousands of devices to be interconnected, which is crucial for applications such as smart cities, smart homes, and industrial automation that require many sensors and actuators to seamlessly connect and communicate. Existing solutions for large-scale MTCD handover authentication in 5G networks have certain security flaws and are almost ineffective in mitigating DoS or DDoS attacks. Moreover, secure functions like bilinear mapping can lead to high computational overheads. All the abovementioned facts motivate us to design a blockchain-assisted group handover authentication protocol to provide sufficient attack prevention, energy efficiency, and fast computation, making it a piece of significant research work. By the BSPGH scheme, the distributed nature of blockchain is utilized to directly achieve mutual authentication between large-scale MTC devices and the target base station, effectively alleviating DoS or DDoS attacks, while also solving the key escrow problem. In terms of resource consumption, the proposed group handover authentication scheme can reduce signaling costs and authentication costs. Our future research will integrate blockchain technology with future communication network standards and protocols to design lightweight protocols.

3. Preliminaries

In this section, we introduce some of the technical concepts and cryptographic techniques that will be used in this paper.

3.1. RelCLAS

Aggregated signatures are a digital signature technique that efficiently combines n independent signatures from n users into a single compact signature. This approach allows verifiers to ensure that these n users have indeed signed their respective n messages, effectively reducing the computational and communication burden during the verification process. In this way, aggregated signatures not only improve data processing efficiency but also optimize resource consumption during storage and transmission. The RelCLAS scheme proposed in [18] is employed in this paper. The RelCLAS scheme typically consists of the following steps:

Setup: The AMF and AUSF receive security parameters to generate the system master keys P_{pub} and T_{pub}, and the system parameters list params is published by the AMF.

Secret Value Generation: Users randomly select $msk_{\text{i}} \in Z_q^*$ as the secret value and compute the user's partial public key mpk_{i}.

Pseudonym Generation: After receiving the user's identity identifier ID, AUSF performs an XOR operation on the ID to obtain the anonymous identity TID.

Partial Secret Key Generation: AMF randomly selects $r_i \in Z_q^*$ as the secret value and generates the user's partial public key R_i and partial private key psk_{i}.

User Key Generation: After receiving the partial key from AMF, the user generates the key pairs $\{mpk_{\text{i}}, R_{\text{i}}\}$ and $\{msk_{\text{i}}, psk_{\text{i}}\}$.

Signature Generation: Each user selects $n_i \in Z_q^*$ to generate the secret value N_i. Using the parameter list *param*, some state information, message $M_i \in M$ (where M is the message space), their anonymous identity TID_i, and their private key pair $\{msk_i, psk_i\}$, they compute the signature σ_i.

Aggregate: The aggregate signature generator is the first user entering a new coverage area. It receives signatures from other users and aggregates these signatures to produce the aggregate signature σ.

Aggregate Verify: The aggregate signature verifier, i.e., t-gNB, uses the anonymous identities TID_i of n users, their corresponding public keys pk_i, secret values N_i, the system master key $Ppub$, and the aggregate signature σ on messages $M_1, \ldots, M_n$ as input. If the aggregate signature is valid, it outputs true, otherwise, it outputs false.

3.2. Blockchain

Blockchain is a collaborative distributed ledger that utilizes multiple computer hosts/nodes in a network to store and manage transaction data. Each host maintains a complete copy of the ledger, eliminating the necessity for a single central authority. Transaction data are organized in chronological order into blocks, with each block containing a certain number of transaction records, typically linked to the previous block utilizing a hash value, forming a chain. It ensures the immutability of transaction data. To ensure the ledger remains consistent across all hosts, the blockchain network uses a consensus algorithm to determine which hosts have the authority to add new blocks. This prevents malicious hosts from tampering with ledger data. Blockchain employs encryption technology to safeguard the confidentiality and integrity of transaction data. Each transaction undergoes a digital signature.

Blockchains are categorized into three types [19]. Public blockchains are open allowing any user to join the blockchain network, view the ledger, and participate in the consensus process. Private blockchains are usually controlled by specific entities or organizations, and only invited participants can join the blockchain network. Consortium blockchains are managed collaboratively by multiple entities or organizations, allowing these participants to share data and jointly manage the blockchain network.

In the system under the study, in normal circumstances, around a small cell controlled by a base station, there are six adjacent cells designated as neighbors. The base station of this cell and those of the adjacent cells are used as nodes in a blockchain network. Each 5G base station in the network serves as a private blockchain node, responsible for storing the public key information of MTCDs. Each base station maintains a complete copy of the blockchain. After an MTCD completes initial registration at a nearby base station, the source base station adds the MTCD's public key information to the blockchain by creating a new transaction. To reduce storage overhead, each block contains multiple transaction records. These transactions are verified by nodes in the blockchain network and consensus can be achieved with other nodes using the Practical Byzantine Fault Tolerance (PBFT) consensus algorithm. Once a new block is accepted by the other blockchain nodes and added to the blockchain, it is distributed to all nodes, including the target base station, thereby updating their blockchain copies. When an MTCD moves from a location controlled by the source base station to the location controlled by the target base station, during the handover preparation phase, the target base station will search for the MTCD's public key information on the entire blockchain.

This blockchain consists of numerous blocks, each with a size of 1MB. Each block is stored as a file containing multiple transaction records, each of which includes the public key information of an MTCD. Given that we use public keys based on the elliptic curve secp256k1 with public keys of 256 bits in length according to [20]. Together with other transaction information including transaction inputs, outputs, version, and other fields, and transaction fees, the total comes to approximately 180 Bytes per transaction. Therefore, using Equation (1), we can calculate how many transactions can be stored in a block. Since

the size of a block is primarily composed of transaction data, for simplicity, we omit other block information in this calculation.

$$N = \frac{M}{P} \tag{1}$$

where N represents the number of users that the base station can serve, M is the block size, and P is the size of the transaction data; the number of transactions that can be stored reaches into the thousands. Considering that the number of MTCDs under a single base station's coverage is only in the dozens, the capacity of the blockchain to store MTCD public key information far exceeds the needs of a group of MTCDs.

4. System Background

4.1. System Model

The system under study follows the structure of the 5G wireless cellular network specified in the 3GPP TS 23.501 R16 [21]. As depicted in Figure 1, the 5G wireless network consists of a core network (CN) and radio access networks (RANs). The devices involved in the CN mainly include the Access and Mobility Management Function (AMF), Authentication Server Function (AUSF), and Unified Data Management (UDM), while Next Generation Node B (gNB) and user equipment or MTCDs exist in the RANs. The entities primarily involved in the handover process are MTCDs, gNB, AMF, AUSF, and UDM.

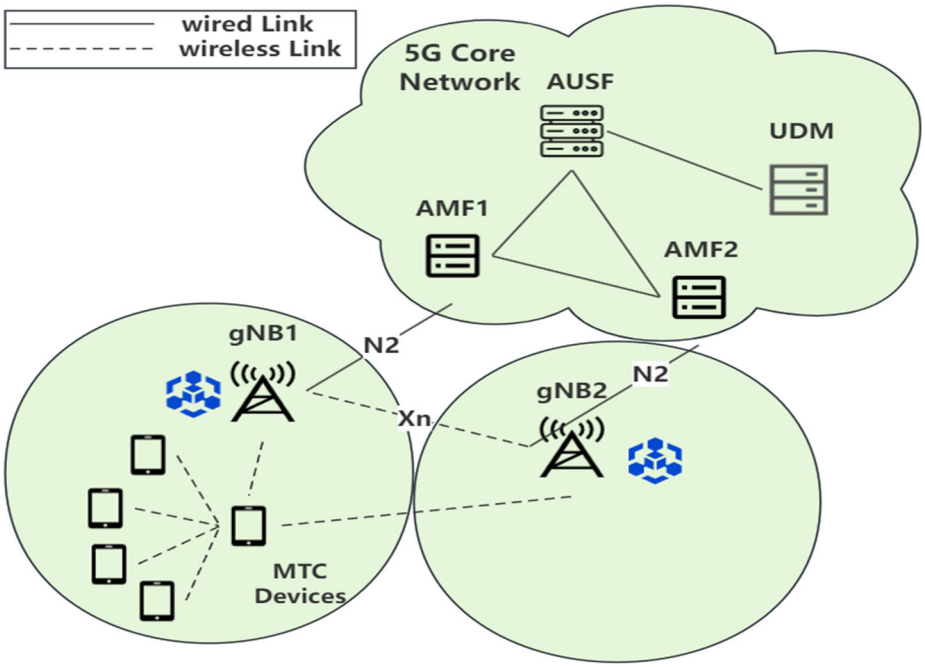

Figure 1. System model.

AMF: AMF is responsible for managing access and mobility-related tasks of user devices, ensuring network efficiency, security, and reliability. In the system, in our group handover authentication phase, the AMF does not need to forward authentication information anymore.

AUSF: AUSF is responsible for handling authentication and security-related tasks for MTCDs. It verifies the identity of MTCDs and the security credentials they provide. In our system, the AUSF primarily generates anonymous identities for both MTCDs and gNBs.

UDM: UDM is responsible for managing and accessing user data. In our system, the UDM stores the permanent identity identifiers of MTCDs and gNBs.

MTCD: In our system, an MTCD, which is a user in the 5G network, initially needs to send a request for registering its identity to the AMF. The MTCD undergoes identity authentication with the gNB upon accessing the network.

gNB: gNB is the base station in the 5G wireless network. It is responsible for MTCD's access and connection, as well as reasonably allocating wireless communication resources.

Blockchain: Blockchain is a distributed database that ensures secure storage and sharing of data by creating a continuously growing and tamper-resistant chain of data records. In our system, a private blockchain is utilized as the secure data structure consisting of registered MTCDs. Each gNB maintains a backup copy of the blockchain.

In the 5G wireless networks specified by the 3GPP standards, when an MTCD enters the signal range of a gNB, it sends an authentication request to the gNB. This request is then forwarded by the gNB to the AMF, which is in turn forwarded to the AUSF. After the AUSF retrieves the necessary key information from the UDM, it executes the authentication process and returns the authentication response to the MTCD through the AMF and the gNB, completing further authentication procedures. In a typical 5G core network, one AUSF typically serves multiple AMFs, and each AMF manages connections with multiple gNBs. By the proposed scheme, decentralization of the authentication entities is achieved by integrating blockchain technology with the 5G system model specified by the 3GPP standard. In this system, each gNB is part of a private blockchain network and holds a copy of the entire blockchain. Once an MTCD completes the initial registration, there is no need to forward the authentication requests to the AUSF and UDM again. When an MTCD needs to undergo a handover authentication, the AMF only needs to forward the information of the service-gNB (s-gNB) to the target-gNB (t-gNB). At this point, the gNB can directly verify the identity of the MTCD using the information stored on the blockchain, without further forwarding to the AMF, AUSF, or UDM. To simplify the design, the proposed scheme focuses on the most common scenario, where all MTCDs connect to the gNB in their home network via 3GPP standard access technologies. Moreover, the connection between the gNB and the 5G core network is provided by a wired connection that is safeguarded by an Internet Protocol Security (IPSec) tunnel. If the MTCD and the gNB successfully achieve mutual authentication, the MTCD can be considered to have secure access to the legitimate 5G network.

In a cellular network architecture, to effectively reuse frequency resources, the entire service area is divided into numerous cells shaped as regular polygons, such as hexagons. Consequently, around a cell controlled by a gNB, typically six adjacent cells are designated as neighbors. During a handover, an MTCD can only hand over to one of these six neighboring cells, which is the cell controlled by the t-gNB associated with the current cell controlled by the s-gNB [22].

4.2. Attack Model

The attack model for the network under study is the Dolev–Yao model [23], by which attackers are assumed to be rational, powerful, and fully controlled entities capable of intercepting, tampering with, and sending messages. They can also attempt to break the security function of the protocol by analyzing communication content. In the RAN domain of the 5G network, there are security vulnerabilities in the wireless communication between MTCDs and gNBs, making them prone to malicious attacks such as information interception and tampering. According to the specification [21], the N2 interface employs IPsec and IKEv2 certificates to protect communications, ensuring their integrity and confidentiality, and preventing replay attacks. Therefore, the connection between the 5G core network and the gNB is considered secure. However, the connection between MTCDs and gNBs is weaker and potentially insecure. It is assumed that the interior of the 5G core network and its network functions are secure, ensuring the safety of connections between network functions. In contrast, other entities in the RAN are not completely trusted.

Given these considerations, an ideal 5G identity handover authentication protocol should support security features including device anonymity, bidirectional identity verifi-

cation, secure data transmission, and perfect forward secrecy. Additionally, it should be capable of defending against active attacks, including impersonation, linkability attacks, replay attacks, man-in-the-middle attacks, and DoS attacks, as well as passive attacks like eavesdropping and location tracking.

5. The Proposed BSPGH

The details of the BSPGH protocol are presented in this section. By combining group authentication with blockchain technology, the BSPGH scheme demonstrates various security attributes. The BSPGH scheme operates in five distinct phases including system initialization and registration, handover preparation, first MTCD handover authentication, group handover authentication, and connection establishment. The notations used are shown in Table 1.

Table 1. Notations and definition of the proposed protocol.

Notation	Definition
TID	Temporary anonymous identity
ID	Permanent identifier
GID	Group key material
MAC	Message authentication code
$H(msg)$	Hash function
TS	Timestamp
K_{it}/K_{1t}	The session key of MTCD and gNB
G	Cyclic additive group
P/q	Generator of the group/prime order of group
PK/SK	Public key/private key
$\{x\}_k$	Encrypted x with key k

5.1. System Initialization and Registration

(1) System initialization: Given security parameter 1^K, AMF selects a large prime number q and $E(F_q)$ as the elliptic curve over a finite field F_q. Let G be a cyclic additive group generated by generator P with order q. $H_0 = Z_q \times \{0,1\}^* \rightarrow Z_q$, $H_1 : G \times \{0,1\}^* \rightarrow Z_q$, $H_2 : \{0,1\}^* \times G \times \{0,1\}^* \times G \times G \rightarrow Z_q$, $H_3 : \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \times \{0,1\}^* \times G \times G \rightarrow Z_q$ and $H_4 : \{0,1\}^* \times \{0,1\}^* \times G \rightarrow Z_q$ are hash functions. AUSF chooses a random element $\alpha \in Z_q$ and calculates the corresponding public key $P_{pub} = \alpha \cdot P$. AMF chooses a random element $\beta \in Z_q$ and calculates the corresponding public key $T_{pub} = \beta \cdot P$. Finally, the AMF publishes the system parameter $params = \left(P, q, G, P_{pub}, T_{pub}, H_0, H_1, H_2, H_3, H_4\right)$.

(2) Initial registration: To protect identity privacy, each MTCD and gNB should first register with the AUSF to obtain their own pseudonyms. Below, we use the MTCD with the real identity ID_{i} as an example to explain the specific registration process, which is shown in Figure 2.

$$Step\text{-}1 : MTCD \rightarrow AMF : \{mpk_i, \{ID_i, mpk_i\}_\delta\}$$

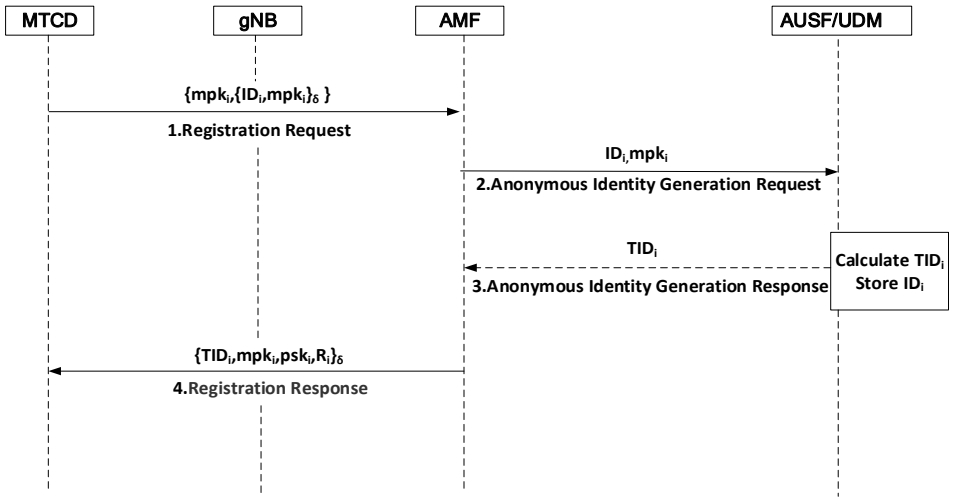

Figure 2. Initial registration.

The MTCD generates a secret value and a corresponding partial public key. It randomly selects a random number $msk_i \in Z_q$ as its secret value and calculates the partial public key $mpk_i = msk_i \cdot P$. Then, it computes $\delta = H_0\left(msk_i \cdot T_{pub}\right)$ as the symmetric encryption key and sends its identity ID_{i} in a message to AMF.

$$Step\text{-}2 : AMF \rightarrow AUSF/UDM : \{ID_i, mpk_i\}$$

AMF checks the identity of $MTCD$ and forwards that identity and the public key to AUSF. The anonymous identity ID is generated by the AUSF, while the UDM stores the permanent identity ID.

$$Step\text{-}3 : AUSF/UDM \rightarrow AMF : \{TID_i\}$$

Upon receiving ID_{i}, AUSF calculates the temporary anonymous identity $TID = ID \oplus H_1(\alpha \cdot mpk_i)$. AUSF then forwards TID_i to AMF. Simultaneously, AUSF also forwards the identity ID_{i} to UDM, which is responsible for storing ID_{i}.

$$Step\text{-}4 : AMF \rightarrow MTCD : \{TID_i, mpk_i, psk_i, R_i\}_\delta$$

AMF generates the secret value and a corresponding partial public key for MTCD. It randomly selects a random number $r_i \in Z_q$, calculates $R_i = r_i \cdot P$, $h_{i1} = H_2\left(TID_i, mpk_i, R_i, T_{pub}\right)$, and $psk_i = r_i + \beta \cdot h_{i1}$, then computes $\delta = H_0(\beta \cdot mpk_i)$. After forwarding the message to $MTCD_i$, $MTCD_i$ stores TID_i, calculates h_{i1}, and verifies that $psk_i \cdot p = R_i + h_{i1} \cdot T_{pub}$. It then sets $PK_i = \{mpk_i, R_i\}$ and $SK_i = \{msk_i, psk_i\}$ as the public–private key pair. The AMF sends the public key to the gNB, and the gNB uploads the public key pair to the blockchain.

(3) Initial authentication: All MTCDs, the AUSF, and the UDM perform the initial authentication following the 5G-AKA scheme. The gNB and AMF monitor the movement trajectory of each MTCD to determine if some MTCDs could form a group based on the grouping algorithm described in [24] that supports the mathematical correlation required to form an MTCD group. If such a correlation is found, these MTCDs will be considered as a group.

5.2. Handover Preparation

This phase occurs before the handover, preparing the necessary key materials for the first MTCD handover authentication and group handover authentication. This phase is shown in Figure 3.

$$Step\text{-}1 : s\text{-}gNB \rightarrow AMF : \{TID_s\}$$

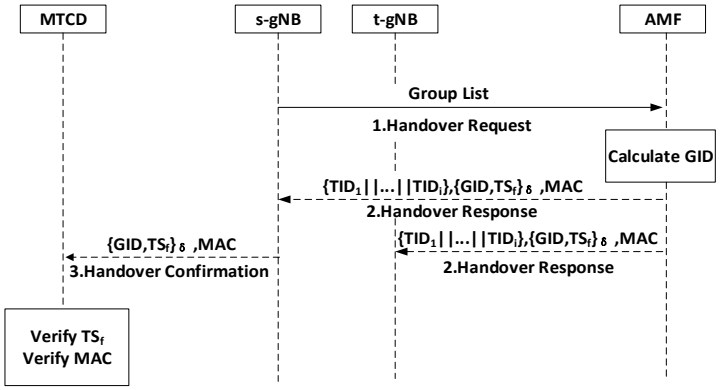

Figure 3. Handover preparation.

The s-gNB sends a handover request to the AMF containing the neighboring gNB list and the temporary identity identifiers $TIDs$ of all group members.

$$Step\text{-}2 : AMF \rightarrow s\text{-}gNB/t\text{-}gNB : \left\{\{TID_1||\ldots||TID_i\}, \left\{GID, TS_f\right\}_\delta, MAC\right\}$$

After receiving all TIDs from the group members, the AMF computes the group key $GID = H(\sum_{i=1}^{n} TID)$. The AMF encrypts the timestamp TS_f and generates the MAC as $MAC = H\left(GID\middle||TS_f\right)$. This message is then broadcasted to the base stations. Upon receiving the message, the t-gNB queries the public key information of MTCDs stored in the blockchain within the group and stores the queried information locally.

$$Step\text{-}3 : s\text{-}gNB \rightarrow \text{MTCD} : \left\{GID, TS_f\right\}_\delta, MAC\}$$

s-gNB broadcasts the message to all MTCDs within the group. Each MTCD decrypts the message to obtain the GID and verifies the received message's timestamps and MAC to determine its authenticity.

5.3. First MTCD Handover Authentication

This phase involves one handover authentication that occurs when the first MTCD enters the range of the t-gNB. After the handover authentication of the first MTCD is successful, subsequent group handover authentication processes will be carried out. This phase is shown in Figure 4.

$$Step\text{-}1 : MTCD \rightarrow s\text{-}gNB : \{TID_1, TS_1, n_1{\cdot}P, Sig_1\}$$

When the first MTCD enters the coverage range of t-gNB within the group, $MTCD_1$ selects a random number $n_1 \in Z_q$ and computes its signature as $Sig_1 = h_{13}msk_1 + n_1 + h_{12}psk_1$. Where $h_{12} = H_3(TID_1, TS_1, PK_1, n_1{\cdot}P)$ and $h_{13} = H_4(TID_1, n_1{\cdot}P)$. Afterwards, $MTCD_1$ sends its TID_1, timestamp TS_1 and $n_1{\cdot}P$, and Sig_1 to s-gNB.

$$Step\text{-}2 : s\text{-}gNB \rightarrow s\text{-AMF} : \{TID_1, TS_1, n_1{\cdot}p, Sig_1\}$$

s-gNB checks the timestamp in the received message and forwards the message to s-AMF.

$$Step\text{-}3 : s\text{-AMF} \rightarrow t\text{-}gNB : \{GID, TS_1, (TID_1||\ldots||TID_\mathrm{i}, n_1{\cdot}P, Sig_1\}$$

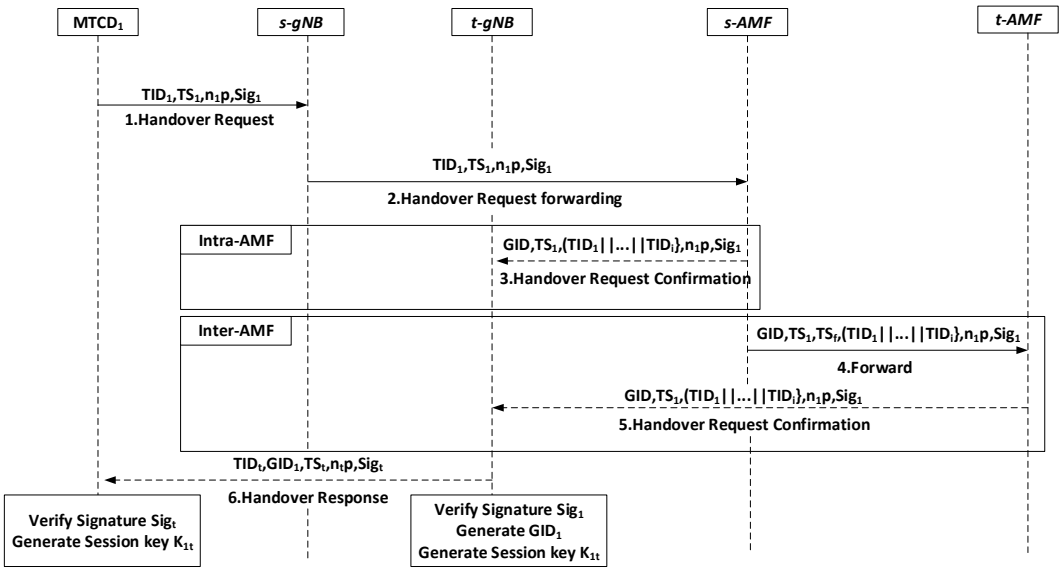

Figure 4. First MTCD handover authentication.

When the handover authentication occurs within the same AMF, after receiving the handover request, s-AMF responds by sending the temporary identity identifiers $TID_1 \ldots TID_{\mathrm{i}}$ of all group members, along with the GID as the group key material, and forwards $n_1 \cdot P$ and Sig_1 to t-gNB. And after receiving the handover response from s-AMF, t-gNB queries the public key of $MTCD_1$ on the blockchain and calculates $h_{11} = H_2\Big(TID_1, mpk_1, R_1, T_{pub}\Big)$, $h_{12} = H_3(TID_1, TS_1, PK_1, n_1 \cdot P)$, and $h_{13} = H_4(TID_1, n_1 \cdot P)$. It verifies the signature $Sig_1 \cdot P = h_{13} \cdot mpk_1 + n_1 \cdot P + h_{12} \cdot R_1 + h_{11} \cdot h_{12} \cdot T_{pub}$, and if the equation holds, $MTCD_1$'s signature is successfully verified. t-gNB then selects a random number $n_t \in Z_q$ and generates the session key $K_{1t} = n_1 \cdot n_t \cdot P$. It also creates the signature $Sig_t = H_0(GID, TID_{\mathrm{t}}, TS_t, n_t \cdot P)$ and computes $GID_1 = \sum_{i=2}^{n} H(TID_{\mathrm{i}})$ as the group key material to authenticate $MTCD_1$. It then sends a message to $MTCD_1$.

$$Step\text{-}4 : s\text{-AMF} \rightarrow t\text{-AMF} : \{GID, TS_1, TS_{\mathrm{f}}, (TID_1 || \ldots || TID_{\mathrm{i}}, n_1 \cdot P, Sig_1\}$$

When the handover authentication occurs between different AMFs, the s-AMF forwards the message to the t-AMF.

$$Step\text{-}5 : t\text{-AMF} \rightarrow t\text{-}gNB : \{GID, TS_1, (TID_1 || \ldots || TID_{\mathrm{i}}, n_1 \cdot P, Sig_1\}$$

Similarly, upon receiving the forwarded handover request, the t-AMF sends the temporary identity identifiers $TID_1 \ldots TID_i$ of all group members, along with the GID as group key material, in response. It then forwards $n_1 \cdot P$ and Sig_1 to t-gNB. Upon receiving the message, the t-gNB proceeds with the verification.

$$Step\text{-}6 : t\text{-}gNB \rightarrow MTCD : \{TID_{\mathrm{t}}, GID_1, TS_t, n_t \cdot P, Sig_t\}$$

After receiving the message from t-gNB, $MTCD_1$ generates the signature $Sig_t' = H_0(GID, TID_{\mathrm{t}}, TS_t, n_t \cdot P)$. If $Sig_t' = Sig_t$, the identity verification of t-gNB is successful. Then, $MTCD_1$ generates the session key $K_{1t} = n_1 \cdot n_t \cdot P$. At this point, the mutual authentication and key negotiation between $MTCD_1$ and t-gNB are complete.

5.4. Group Handover Authentication

$MTCD_1$ initiates an aggregated signature request to other group members in the group, broadcasting the temporary identity TID_{t} of the t-gNB. $MTCD_1$ also sends its

temporary identity TID_1 and the group key material GID_1. Other members within the group verify the identity of $MTCD_1$ by validating $GID = GID_1 + H(TID_1)$.

$$Step\text{-}1 : MTCD_1 \rightarrow MTCD_i : \{GID_1, TID_t, TID_1, TS_1\}$$

$$Step\text{-}2 : MTCD_i \rightarrow MTCD_1 : \{TID_\mathrm{i}, TS_\mathrm{i}, n_i{\cdot}P, Sig_i\}$$

The group member $MTCD_\mathrm{i}$ receives the aggregated signature request, verifies the identity of $MTCD_1$, selects a random number $n_\mathrm{i} \in Z_\mathrm{q}$, generates its respective signatures, and sends its signature $Sig_i = h_{i3}msk_i + n_i + h_{i2}psk_i$.

$$Step\text{-}3 : MTCD_1 \rightarrow t\text{-}gNB : \{(TID_2||\ldots||TID_i), (n_2{\cdot}P||\ldots||n_i{\cdot}P), Sig, TS_i\}$$

After receiving the signatures from all group members, $MTCD_1$ calculates the aggregated signature $Sig = \sum_{i=2}^{n} Sig_\mathrm{i}$ for the group and sends the aggregated signature to the t-gNB for performing group signature verification using the equation $\sum_{i=2}^{n} Sig_i{\cdot}P = \sum_{i=2}^{n} h_{i3}{\cdot}mpk_i + \sum_{i=2}^{n} n_i \cdot P + \sum_{i=2}^{n} h_{i2}{\cdot}R_i + \sum_{i=2}^{n} h_{i1}{\cdot}h_{i2}{\cdot}T_{pub}$ to pre-authenticate all group members. If the equation holds, the t-gNB verifies all MTCDs. Subsequently, t-gNB generates the session key $K_{it} = n_i{\cdot}n_t{\cdot}P$ between $MTCD_\mathrm{i}$ and the t-gNB.

$$Step\text{-}4 : t\text{-}gNB \rightarrow MTCD_1 : \{TID_t, TS_t, n_t{\cdot}P, Sig_t\}$$

The t-gNB verifies the signatures of the group members to authenticate their identities. If the identity authentication is successful, the t-gNB sends the timestamp TS_t, random number $n_\mathrm{t}{\cdot}P$, and its own signature $Sig_t = H_0(GID, TID_\mathrm{t}, TS_t, n_t{\cdot}P)$ to the $MTCD_1$, because it has passed the first MTCD handover authentication already.

$$Step\text{-}5 : MTCD_1 \rightarrow MTCD_i : \{TID_\mathrm{t}, TS_1, TS_\mathrm{t}, n_t{\cdot}P, Sig_t\}$$

After receiving the message from $MTCD_1$, $MTCD_\mathrm{i}$ generates the signature $Sig'_t = H_0(GID, TID_\mathrm{t}, TS_t, n_t{\cdot}P)$. If $Sig'_t = Sig_t$, the identity verification of t-gNB is successful. If the verification is successful, $MTCD_\mathrm{i}$ generates the session key $K_{it} = n_i{\cdot}n_t{\cdot}P$ between $MTCD_\mathrm{i}$ and the t-gNB. At this point, mutual authentication and key negotiation between $MTCD_\mathrm{i}$ and t-gNB are complete. This phase is shown in Figure 5.

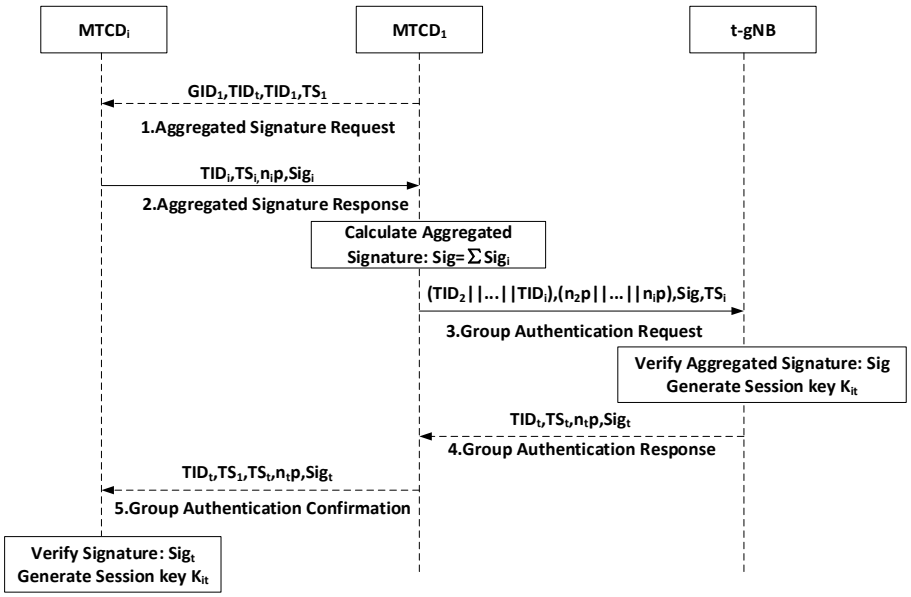

Figure 5. Group handover authentication.

6. Security Evaluation

In this section, we first prove the logic correctness of the proposed BSPGH scheme by using BAN logic and perform a formal verification of the security functionality of the BSPGH protocol by using the Scyther. Furthermore, a security analysis is conducted to identify security properties held by the BSPGH protocol and its robustness against various malicious attacks is described. The results can provide insights into the protocol's effectiveness and capacity to resist various threats.

6.1. Formal Proof by BAN Logic

BAN logic is an important method for the formal analysis of security protocols [25], aiming to verify their security and correctness. The fundamental principle of BAN logic is to establish a set of rigorous logical rules to describe the semantics of entities, message exchange, and information states involved in the protocol. To apply BAN logic to prove the BSPGH protocol, we formalize the protocol into an idealized form. We then propose assumptions and objectives and use BAN logic symbols and rules for derivation, such as message meaning rules, temporary value validation rules, arbitration rules, belief rules, and reception rules. By manually applying the derivation rules, we aim to achieve the objectives and verify the security properties of the protocol.

The rules of BAN logic for derivation can be described as follows:

(R1) The Message Meaning Rule: $\frac{P|\equiv Q\overset{K}{\longleftrightarrow}P, P\triangleleft X_K}{P|\equiv Q|\sim X}$, $\frac{P|\equiv\overset{K}{\rightarrow}Q, P\triangleleft X_{K^{-1}}}{P|\equiv Q\sim X}$, $\frac{P|\equiv P\overset{Y}{\rightleftharpoons}Q, P\triangleleft X_Y}{P|\equiv Q\sim X}$. The first means that if party P trusts that K is a shared key between P and Q, and if P has received a message X encrypted with K before, then P believes that Q has sent the message X. The second means that if P believes that user Q's public key is K, and P sees that the message X, which is signed with Q's private key, is K^{-1}, then P believes that the message X was sent by Q. The third means describes the shared secret.

(R2) The Freshness Rule: $\frac{P|\equiv\#(X)}{P|\equiv\#(X,Y)}$. This rule means that if one part of the message is fresh, then the entire message is fresh.

(R3) The Nonce Verification Rule: $\frac{P|\equiv\#(X), P|\equiv Q|\sim X}{P|\equiv Q|\equiv X}$. This rule means that if P believes that message X is fresh and believes that Q has sent X before, then P believes that Q believes X.

(R4) The Belief Conjunction Rule: $\frac{P|\equiv Q|\equiv(X,Y)}{P|\equiv Q|\equiv X}$, $\frac{P|\equiv X, P|\equiv Y}{P|\equiv(X,Y)}$, $\frac{P|\equiv(X,Y)}{P|\equiv X}$. This rule means that if P believes that party Q believes messages X and Y, then P believes that Q believes X.

(R5) The Jurisdiction Rule: $\frac{P|\equiv Q|\Rightarrow X, P|\equiv Q|\equiv X}{P|\equiv X}$. This rule means that if P believes Q has jurisdiction on message X, and P believes Q believes X, then P believes X.

6.1.1. Formalized Protocol

To idealize the protocol, we describe the messages in the proposed protocol as follows:
Messages-1: The s-gNB sends $\{TIDs\}$ to AMF.
(M1) $AMF \triangleleft TIDs$
Messages-2: The AMF sends $\{(TID_1||\ldots||TID_{\mathrm{i}}), \{GID, TS_{\mathrm{f}}\}_{\delta}, MAC\}$ to s-gNB.
(M2) $s\text{-}gNB \triangleleft \{\{TID_1||\ldots||TID_{\mathrm{i}}\}, \{GID, TS_{\mathrm{f}}\}_{\delta}, MAC\}$
Messages-3: The s-gNB sends $\{GID, TS_{\mathrm{f}}\}_{\delta}, MAC$ to MTCD.
(M3) $MTCD \triangleleft \{\{GID, TS_{\mathrm{f}}\}_{\delta}, MAC\}$
Messages-4: The $MTCD_1$ sends $\{TID_1, TS_1, n_1{\cdot}p, Sig_1\}$ to s-gNB.
(M4) $s\text{-}gNB \triangleleft \{TID_1, TS_1, n_1{\cdot}p, Sig_1\}_{PK^{-1}}$
Messages-5: The s-gNB sends $\{TID_1, n_1{\cdot}p, Sig_1\}$ to AMF.
(M5) $AMF \triangleleft \{TID_1, n_1{\cdot}p, Sig_1\}_{PK^{-1}}$
Messages-6: The AMF sends $\{GID, TS_1, (TID_1||\ldots||TID_{\mathrm{i}}), n_1{\cdot}p, Sig_1\}$ to t-gNB.
(M6) $t\text{-}gNB \triangleleft \{GID, TS_1, (TID_1||\ldots||TID_{\mathrm{i}}), n_1{\cdot}p, Sig_1\}_{PK^{-1}}$
Messages-7: The t-gNB sends $\{TID_{\mathrm{t}}, GID_1, TS_{\mathrm{t}}, n_t{\cdot}p, Sig_t\}$ to $MTCD_1$.
(M7) $MTCD_1 \triangleleft \{TID_{\mathrm{t}}, GID_1, TS_{\mathrm{t}}, n_t{\cdot}p, Sig_t\}_{GID}$
Messages-8: The $MTCD_1$ sends $\{GID_1, TID_{\mathrm{t}}, TID_1, TS_1\}$ to $MTCD_{\mathrm{i}}$.

(M8) $MTCD_i \triangleleft \{GID_1, TID_t, TID_1, TS_1\}_{GID}$
Messages-9: The $MTCD_i$ sends $\{TID_i, TS_i, n_i{\cdot}p, Sig_i\}$ to $MTCD_1$.
(M9) $MTCD_1 \triangleleft \{TID_i, TS_i, n_i{\cdot}p, Sig_i\}_{PK^{-1}}$
Messages-10: The $MTCD_1$ sends $\{(TID_2||\ldots||TID_i), n_2{\cdot}p||\ldots||n_ip, Sig, TS_i\}$ to t-gNB.
(M10) $t\text{-}gNB \triangleleft \{(TID_2||\ldots||TID_i), n_2{\cdot}p||\ldots||n_i{\cdot}p, Sig, TS_i\}_{PK^{-1}}$
Messages-11: The t-gNB sends $\{TID_t, TS_t, n_t{\cdot}p, Sig_t\}$ to $MTCD_1$.
(M11) $MTCD_1 \triangleleft \{TID_t, TS_t, n_t{\cdot}p, Sig_t\}_{GID}$
Messages-12: The $MTCD_1$ sends $\{TID_t, TS_1, TS_t, n_t{\cdot}p, Sig_t\}$ to $MTCD_i$.
(M12) $MTCD_i \triangleleft \{TID_t, TS_1, TS_t, n_t{\cdot}p, Sig_t\}_{GID}$
We refer to all MTC devices, including MTCD and $MTCD_i$, as MTCD.

6.1.2. Logical Assumptions

The initial assumptions for protocol analysis are as follows: MTCDs and t-gNB trust locally generated random numbers, as well as the key pairs generated from these random numbers. The random number n includes both n_1 and n_i, and the MTCDs include both $MTCD_1$ and $MTCD_i$.

(A1) $MTCD_1|\equiv n_1$, $MTCD_i| \equiv n_i$
(A2) $MTCD_1| \equiv n_1{\cdot}p$, $MTCD_i| \equiv n_i{\cdot}p$
(A3) $t\text{-}gNB| \equiv n_t$
(A4) $t\text{-}gNB| \equiv n_t{\cdot}p$

Both MTCDs and t-gNB, upon receiving messages, verify the timestamps. Therefore, they trust the freshness of the timestamps.

(A5) $MTCD| \equiv \#(TS_t)$
(A6) $t\text{-}gNB| \equiv \#(TS_1)$, $t\text{-}gNB| \equiv \#(TS_i)$

MTCDs should trust that the keys generated by t-gNB are under its control and trustworthy. Similarly, t-gNB should trust that the keys generated by $MTCD_i$ are under its control and trustworthy. This mutual trust is established because $MTCD_i$ undergoes the initial handover authentication process. And upon successful authentication, it gains complete trust from t-gNB. Consequently, $MTCD_i$ should also trust that the keys from t-gNB received by $MTCD_i$ are under control and trustworthy.

(A7) $MTCD| \equiv t\text{-}gNB \Rightarrow n_t{\cdot}p$
(A8) $t\text{-}gNB| \equiv MTCD_1 \Rightarrow n_1{\cdot}p$, $t\text{-}gNB| \equiv MTCD_i \Rightarrow n_i{\cdot}p$
(A9) $MTCD|\equiv t\text{-}gNB| \equiv n_t$
(A10) $t\text{-}gNB|\equiv MTCD_1| \equiv n_1$, $t\text{-}gNB|\equiv MTCD_i| \equiv n_i$

Both t-gNB and MTCDs possess the same group key. The group key is generated by MTCDs in the handover process and then sent to t-gNB. Both parties trust this key.

(A11) $MTCD| \equiv MTCD \overset{GID}{\longleftrightarrow} t\text{-}gNB$
(A12) $t\text{-}gNB| \equiv \overset{PK}{\rightarrow} MTCD$
The group key GID is the shared key among the group of MTCDs.

6.1.3. Protocol Goal

The purpose of the handover authentication is to accomplish mutual authentication and key agreement between each MTCD and the t-gNB. Since the first $MTCD_1$ and subsequent $MTCD_i$ have different authentication processes, we will prove them separately. In the model, all MTCDs, including $MTCD_1$ and $MTCD_i$, are represented as MTCD, and the timestamp is denoted as TS. The specific objectives are described as follows:

G1–G8 are objectives. If all 8 objectives are achieved, then the session key, which is known only to MTCD and t-gNB, will be shared between them.

(G1) $MTCD_1| \equiv MTCD_1 \overset{K_{1t}}{\longleftrightarrow} t\text{-}gNB$
(G2) $t\text{-}gNB| \equiv t\text{-}gNB \overset{K_{1t}}{\longleftrightarrow} MTCD_1$
(G3) $MTCD_1|\equiv t\text{-}gNB| \equiv t\text{-}gNB \overset{K_{1t}}{\longleftrightarrow} MTCD_1$
(G4) $t\text{-}gNB|\equiv MTCD_1| \equiv MTCD_1 \overset{K_{1t}}{\longleftrightarrow} t\text{-}gNB$

(G5) $MTCD_{\mathrm{i}}| \equiv MTCD_{\mathrm{i}} \overset{K_{\mathrm{it}}}{\longleftrightarrow} t\text{-}gNB$
(G6) $t\text{-}gNB| \equiv t\text{-}gNB \overset{K_{\mathrm{it}}}{\longleftrightarrow} MTCD_{\mathrm{i}}$
(G7) $MTCD_{\mathrm{i}}| \equiv t\text{-}gNB| \equiv t\text{-}gNB \overset{K_{\mathrm{it}}}{\longleftrightarrow} MTCD_{\mathrm{i}}$
(G8) $t\text{-}gNB| \equiv MTCD_{\mathrm{i}}| \equiv MTCD_{\mathrm{i}} \overset{K_{\mathrm{it}}}{\longleftrightarrow} t\text{-}gNB$

6.1.4. Protocol Verification

Using the rules, assumptions, and messages, the detailed proof is as follows:
According to R1 and considering A12 and M6, we can deduce

$$t\text{-}gNB| \equiv MTCD_1 \sim \{GID, TS_1, (TID_1|| \ldots ||TID_{\mathrm{i}}), n_1 \cdot p, Sig_1\} \tag{2}$$

According to R2 and considering A6, we can deduce

$$t\text{-}gNB| \equiv \#\{GID, TS_1, (TID_1|| \ldots ||TID_{\mathrm{i}}), n_1 \cdot p, Sig_1\} \tag{3}$$

According to R3 and considering (2) and (3), we can deduce

$$t\text{-}gNB| \equiv MTCD_1| \equiv \{GID, TS_1, (TID_1|| \ldots ||TID_{\mathrm{i}}), n_1 \cdot p, Sig_1\} \tag{4}$$

According to R4 and considering (4), we can deduce

$$t\text{-}gNB| \equiv MTCD_1| \equiv \{n_1 \cdot p\} \tag{5}$$

According to R5, along with A8 and (5), we can deduce

$$t\text{-}gNB| \equiv \{n_1 \cdot p\} \tag{6}$$

According to the R1 and considering A11 and M7, we can deduce

$$MTCD_1| \equiv t\text{-}gNB \sim \{GID_1, TS_{\mathrm{t}}, n_t \cdot p, Sig_t\} \tag{7}$$

According to R2 and considering A5, we can deduce

$$MTCD_1| \equiv \#\{GID_1, TS_{\mathrm{t}}, n_t \cdot p, Sig_t\} \tag{8}$$

According to R3 and considering (7) and (8), we can deduce

$$MTCD_1| \equiv t\text{-}gNB| \equiv \{GID_1, TS_{\mathrm{t}}, n_t \cdot p, Sig_t\} \tag{9}$$

According to R4 and considering (9), we can deduce

$$MTCD_1| \equiv t\text{-}gNB| \equiv \{n_{\mathrm{t}} \cdot p\} \tag{10}$$

According to R5 and considering (10) and A7, we can deduce

$$MTCD_1| \equiv \{n_{\mathrm{t}} \cdot p\} \tag{11}$$

According to R4 and considering (6) and A3, we can deduce

$$t\text{-}gNB| \equiv \{n_1 \cdot n_{\mathrm{t}} \cdot p\} \tag{12}$$

Since $n_1 \cdot n_t \cdot p$ is the shared key $K_{1\mathrm{t}}$, between $MTCD_1$ and t-gNB; therefore, according to Equation (12), we can deduce:

$$t\text{-}gNB| \equiv t\text{-}gNB \overset{K_{1\mathrm{t}}}{\longleftrightarrow} MTCD_1 \tag{13}$$

According to R4 and considering (11) and A1, we can deduce

$$MTCD_1| \equiv \{n_1 \cdot n_t \cdot p\} \tag{14}$$

As mentioned above, according to (14), we can deduce

$$MTCD_1| \equiv MTCD_1 \overset{K_{1t}}{\longleftrightarrow} t\text{-}gNB \tag{15}$$

Furthermore, to complete the process, t-gNB receives M6 and verifies it. It must trust M6 to proceed with the protocol and send M7. Therefore, if $MTCD_1$ has already received M7, $MTCD_1$ can infer that t-gNB already trusts M6. $n_1 \cdot p$ is included in M6. Therefore, we can deduce

$$MTCD_1| \equiv t\text{-}gNB| \equiv \{n_1 \cdot p\} \tag{16}$$

According to R4 and considering (16) and A9, we can deduce

$$MTCD_1| \equiv t\text{-}gNB| \equiv MTCD_1 \overset{K_{1t}}{\longleftrightarrow} t\text{-}gNB \tag{17}$$

As mentioned above, according to M7 and M8, we can deduce

$$t\text{-}gNB| \equiv MTCD_1| \equiv \{n_t \cdot p\} \tag{18}$$

According to R4 and considering (18) and A10, we can deduce

$$t\text{-}gNB| \equiv MTCD_1| \equiv t\text{-}gNB \overset{K_{1t}}{\longleftrightarrow} MTCD_1 \tag{19}$$

According to R1 and considering M10 and A12, we can deduce

$$t\text{-}gNB| \equiv MTCD_i \sim \{(TID_2|| \ldots ||TID_i), n_2 \cdot p|| \ldots ||n_i \cdot p, Sig, TS_i\} \tag{20}$$

According to R2 and considering A6, we can deduce

$$t\text{-}gNB| \equiv \#\{(TID_2|| \ldots ||TID_i), n_2 \cdot p|| \ldots ||n_i \cdot p, Sig, TS_i\} \tag{21}$$

According to R3 and considering (20) and (21), we can deduce

$$t\text{-}gNB| \equiv MTCD_i| \equiv \{(TID_2|| \ldots ||TID_i), n_2 \cdot p|| \ldots ||n_i \cdot p, Sig, TS_i\} \tag{22}$$

According to R4 and considering (22), we can deduce

$$t\text{-}gNB| \equiv MTCD_i| \equiv \{n_i \cdot p\} \tag{23}$$

According to R5 and considering (23) and A8, we can deduce

$$t\text{-}gNB| \equiv \{n_i \cdot p\} \tag{24}$$

According to R1 and considering M12 and A11, we can deduce

$$MTCD_i| \equiv t\text{-}gNB \sim \{TS_1, TS_t, n_t \cdot p, Sig_t\} \tag{25}$$

According to R2 and considering A5, we can deduce

$$MTCD_i| \equiv \#\{n_t \cdot p, TS_1, Sig_t\} \tag{26}$$

According to R3 and considering (25) and (26), we can deduce

$$MTCD_i| \equiv t\text{-}gNB| \equiv \{n_t \cdot p, TS_1, Sig_t\} \tag{27}$$

According to R4 and considering (27), we can deduce

$$MTCD_i \equiv t\text{-}gNB| \equiv \{n_t \cdot p\} \tag{28}$$

According to R5 and considering (28) and A7, we can deduce

$$MTCD_i| \equiv \{n_t \cdot p\} \tag{29}$$

According to R4 and considering (29) and A1, we can deduce

$$MTCD_i| \equiv \{n_i \cdot n_t \cdot p\} \tag{30}$$

As mentioned above, according to (30), we can deduce

$$MTCD_i| \equiv MTCD_i \overset{K_{it}}{\longleftrightarrow} t\text{-}gNB \tag{31}$$

According to R4 and considering (31) and A3, we can deduce

$$t\text{-}gNB| \equiv \{n_i \cdot n_t \cdot p\} \tag{32}$$

As mentioned above, according to (32), we can deduce

$$t\text{-}gNB| \equiv t\text{-}gNB \overset{K_{it}}{\longleftrightarrow} MTCD_i \tag{33}$$

Similar to $MTCD_1$, t-gNB also receives M10 first, and it must trust M10 in order to proceed with the protocol and send M12. If $MTCD_i$ has already received M12, then $MTCD_i$ can conclude that t-gNB now trusts M10. According to M10 and M12, we can deduce

$$MTCD_i| \equiv t\text{-}gNB| \equiv \{n_i \cdot p\} \tag{34}$$

According to R4 and considering (34) and A9, we can deduce

$$MTCD_i| \equiv t\text{-}gNB| \equiv MTCD_i \overset{K_{it}}{\longleftrightarrow} t\text{-}gNB \tag{35}$$

Likewise, based on M12, after $MTCD_i$ verifies the identity of t-gNB without errors, it generates a session key for subsequent communication. From this, we can deduce

$$t\text{-}gNB| \equiv MTCD_i| \equiv \{n_t \cdot p\} \tag{36}$$

According to R4 and considering (36) and A10, we can deduce

$$t\text{-}gNB| \equiv MTCD_i| \equiv t\text{-}gNB \overset{K_{it}}{\longleftrightarrow} MTCD_i \tag{37}$$

In summary, we have achieved all the security objectives, ensuring key negotiation and mutual authentication in the protocol. Our protocol has been logically validated.

6.2. Formal Verification

Scyther Tool is a formal verification tool commonly used for validating security protocols [26]. Scyther offers four security statements to ensure protocol consistency and detect various attacks like message forgery, replay, and man-in-the-middle (MITM). These include "Aliveness" for completing protocol steps with active responders, "Niagree" for the correct variable receipt without one-to-one communication, "Nisynch" for the expected protocol operation without one-to-one synchronization, and "Weakagree" for one-to-one communication within the same group of initiators or responders.

The verification results are shown in Figure 6, where Figure 6a demonstrates the first MTCD handover authentication phase, and Figure 6b demonstrates the group handover

authentication phase. Specifically, the model is created with three roles: t-gNB, $MTCD_{\mathrm{i}}$, and $MTCD_1$. Initially, the BSPGH scheme achieves mutual key agreement using the Elliptic Curve Diffie–Hellman (ECDH). To simulate the ECDH key exchange between two parties, two functions are defined as $g1$ and $g2$. Then, $n_1 \cdot P$ is set to $g1(n1)$, $n_{\mathrm{i}} \cdot P$ is set to $g1(ni)$, and $n_{\mathrm{t}} \cdot P$ is set to $g1(nt)$. The confidentiality of the ECDH private keys is first verified using the declarations of Secret $n1$, Secret ni, and Secret nt. Next, the derivation of ECDH public keys is performed using $K1t$ (i.e., $g2(nt,\ g1(n1))$) and Kit (i.e., $g2(nt,\ g1(ni))$). The confidentiality of the ECDH private keys is ensured by declaring Secret ni, Secret nt, and Secret $n1$. Additionally, the confidentiality of the ECDH public key is validated by the SKR declaration. Our protocol assumes that $MTCD_{\mathrm{i}}$ needs to use the group key GID to verify the identity of $MTCD_1$. To ensure that adversaries cannot obtain the group key GID in any way, its secrecy is checked using the Secret GID declaration. Finally, the verification results reveal that all participants in the system satisfy the properties of synchronous (Nisynch), consistent (Niagree), active (Alive), and weak consistency (Weakagree). All the keys meet the security requirements. Therefore, our protocol is deemed secure by the verification using Scyther.

Claim				Status		Comments
handover	MTCD1	handover,MTCD11	Secret n1	Ok	Verified	No attacks.
		handover,MTCD12	Secret GID	Ok	Verified	No attacks.
		handover,MTCD13	SKR K1t	Ok	Verified	No attacks.
		handover,MTCD14	Alive	Ok	Verified	No attacks.
		handover,MTCD15	Weakagree	Ok	Verified	No attacks.
		handover,MTCD16	Niagree	Ok	Verified	No attacks.
		handover,MTCD17	Nisynch	Ok	Verified	No attacks.
	tgNB	handover,tgNB1	Secret nt	Ok	Verified	No attacks.
		handover,tgNB2	SKR K1t	Ok	Verified	No attacks.
		handover,tgNB3	Alive	Ok	Verified	No attacks.
		handover,tgNB4	Weakagree	Ok	Verified	No attacks.
		handover,tgNB5	Niagree	Ok	Verified	No attacks.
		handover,tgNB6	Nisynch	Ok	Verified	No attacks.

(**a**) The first MTCD handover authentication

MTCDi	handover,MTCDi1	Secret ni	Ok	Verified	No attacks.	
	handover,MTCDi2	SKR Kit	Ok	Verified	No attacks.	
	handover,MTCDi3	Secret GID	Ok	Verified	No attacks.	
	handover,MTCDi4	Alive	Ok	Verified	No attacks.	
	handover,MTCDi5	Weakagree	Ok	Verified	No attacks.	
	handover,MTCDi6	Niagree	Ok	Verified	No attacks.	
	handover,MTCDi7	Nisynch	Ok	Verified	No attacks.	
MTCD1	handover,MTCD11	Secret n1	Ok	Verified	No attacks.	
	handover,MTCD12	SKR K1t	Ok	Verified	No attacks.	
	handover,MTCD13	Secret GID	Ok	Verified	No attacks.	
	handover,MTCD14	Alive	Ok	Verified	No attacks.	
	handover,MTCD15	Weakagree	Ok	Verified	No attacks.	
	handover,MTCD16	Niagree	Ok	Verified	No attacks.	
	handover,MTCD17	Nisynch	Ok	Verified	No attacks.	
tgNB	handover,tgNB1	Secret nt	Ok	Verified	No attacks.	
	handover,tgNB2	SKR K1t	Ok	Verified	No attacks.	
	handover,tgNB3	SKR Kit	Ok	Verified	No attacks.	
	handover,tgNB4	Alive	Ok	Verified	No attacks.	
	handover,tgNB5	Weakagree	Ok	Verified	No attacks.	
	handover,tgNB6	Niagree	Ok	Verified	No attacks.	
	handover,tgNB7	Nisynch	Ok	Verified	No attacks.	

(**b**) The group handover authentication

Figure 6. Results of formal verification.

6.3. Security Analysis

- **Mutual Authentication:** By the BSPGH scheme, a gNB verifies the authenticity of the MTCD's signature *Sig* by retrieving the public key stored on the blockchain, thereby confirming the identity of the MTCD. Since digital signatures are generated by encrypting messages with a private key and it is computationally infeasible to deduce the private key from the public key, this way allows the MTCD to effectively prove the legitimacy of its authentication request to the gNB. Furthermore, the gNB employs hash-based message authentication code (HMAC) scheme and uses the group key

GID as the key for the HMAC to generate a signature Sig_t. The MTCD, possessing a legitimate GID, verifies Sig_t to prove the legitimacy of the response. In this way, the MTCD and the gNB are able to achieve mutual authentication, ensuring the security of the communication between them.

- **Privacy protection:** The temporary identity identifier TID_i is transmitted to each participant over the wireless channel, ensuring that the real identity is only disclosed to legitimate gNBs and the core network. It satisfies anonymity requirements.
- **Perfect forward secrecy and backward secrecy:** By the proposed protocol, the generation of the session key K_{it} relies on randomly generated ECDH parameters. Without the session key, attackers cannot recover the contents of a specific session. Moreover, since a session key is generated for each session and the ECDH parameters for each session key are independent of those from previous or future sessions, the leakage of a session key would only affect the current session. The confidentiality of previous or future sessions would remain unaffected.
- **Replay and impersonation attacks resistance:** By the proposed protocol, authentication requests and responses are both marked with a timestamp TS. The TS constraint ensures that messages are received within a specified time window, allowing easy identification. The replayed messages will be discarded, thus, countering replay attacks. The use of a key system based on discrete logarithms makes deriving private keys from the public keys challenging, preventing attackers from forging signatures and thus impersonating legitimate identities. It can effectively strengthen the ability of the protocol to resist impersonation attacks.
- **DoS/DDoS attacks resistance:** The receiving gNB first verifies the timestamp's validity and then compares the signature with records in the blockchain for authentication. If they do not match, the session is immediately terminated, preventing attackers from consuming gNB's computational resources through replay attacks. The failure of one gNB or one AMF will not affect the entire 5G wireless network. Therefore, it can prevent DDoS attacks in 5G authentication.
- **Session key leakage:** Attackers may attempt to compute the session key to steal messages transmitted over the wireless channel. However, the session key is formed based on ECDH by the proposed scheme, which relies on the difficulty of the elliptic curve discrete logarithm problem. Attackers cannot obtain n_{i} and n_{t} from $n_{\mathrm{i}} \cdot p$ and $n_{\mathrm{t}} \cdot p$, and thus cannot compute the session key $n_{\mathrm{i}} \cdot n_{\mathrm{t}} \cdot p$. It effectively prevents session key leakage.
- **Sybil attack:** By the proposed protocol, a private blockchain is utilized. Within a private blockchain, access and participation in the network are restricted, allowing only MTCDs that have been registered by core network entities to join and interact. Furthermore, the identities of MTCDs must be verified subsequently, which limits the ability of attackers to forge numerous identities to conduct attacks. Therefore, this approach is capable of resisting Sybil attacks, where an attacker creates a large number of pseudonymous identities to compromise the network.
- **Man-in-the-middle attack:** By the BSPGH scheme, an adversary cannot impersonate a legitimate t-gNB to deceive an MTCD because a temporary session key K_{it} is established between them using the ECDH. The adversary cannot obtain or modify the temporary session key; thus, it is unable to establish communication with the MTCD.
- **Linkability attack prevention:** In the BSPGH authentication process, the PK and TID are periodically updated, while the elements in other messages are random numbers. BSPGH employs ECDH for session key generation instead of using serial numbers, which prevents the common MAC failures or synchronization issues found in symmetric key-based AKA protocols. This approach makes it difficult for attackers to analyze the correlation between different messages or to exploit erroneous messages as vulnerabilities to track a specific device. Consequently, BSPGH effectively safeguards against linkability attacks, enhancing privacy and security in the communication process.

7. Performance Evaluation

In this section, we evaluated the performance of the BSPGH scheme in terms of computational cost and communication cost during the first handover authentication and group handover authentication phases and compare its performance with those of three other protocols, namely, 5G standard specified by 3GPP TS 23.501 R16 [21], a privacy-preserving handover authentication protocol for a group of MTC devices in 5G networks (PPHAP) [1], and a novel authentication scheme supporting multiple user access for 5G and beyond (NASS) [7]. In the analysis, it is assumed that all symmetric encryption keys are 256 bits, $MACs$ are 160 bits, and elements in the Hash functions, TID, GID, Sig, $n \cdot P$, and Z_q^* are 128 bits. The timestamp is represented by 32 bits, and elements in group G have sizes of 320 bits.

7.1. Blockchain Operation Cost

To evaluate the data access latency in the blockchain, we followed the evaluation methodology outlined in [27]. Initially, a blockchain prototype was created. This blockchain comprised numerous blocks, each being 1 MB in size. Every block, stored as a file, contained multiple transaction records, with each transaction being a subscription record of a device. To reduce the storage overhead of the blockchain, this size should be adjusted according to the preferences of the network operators. By default, we followed the Bitcoin model with a block size of 1 MB. All transactions in the blockchain were indexed using Python dictionaries. To compare the performance differences between the blockchain and traditional databases, another centralized database was also constructed using MariaDB v10.4.14 [28]. The read operation $T_{BC.read}$ is 0.2914 ms and the write operation $T_{BC.write}$ is 0.0434 ms, while for the centralized database, both of the read and write operations T_{DB} are 0.4956 ms.

By the proposed scheme, since the writing and querying operations of the blockchain information occur prior to the two phases of handover authentication, specifically during the registration phase and the handover preparation phase, the time taken for blockchain operations has not been included in the computational costs of the first handover authentication and group handover authentication phases.

7.2. Computational Cost

We utilize a device equipped with an Intel(R) Core (TM) i7-12700H processor running at 3.50 GHz and 16 GB of RAMs for the evaluation of the computational cost of the two phases of handover authentication. The cryptographic library employed to perform the required cryptographic operations of the proposed scheme was C/C++ OPENSSL. For the selected protocols, each protocol uses different encryption functions with unique key size requirements. To assess their performance comprehensively and fairly at the same security level, we follow the recommendation of NIST in [28] and use a 256-bit equivalent key strength throughout the entire simulation process. Therefore, for all operations based on ECC, secp256k1 is selected as the default elliptic curve. We run each encryption function 10,000 times to measure its average execution time. The measurement results obtained follow. The point multiplication T_{PM} is 0.201 ms. The point addition T_{PA} is 0.001 ms. The modular exponentiation T_{E} is 0.665 ms. The Rivest–Shamir–Adleman (RSA) signature and verification T_{RV} is 1.445 ms. The hash operation T_{H} is 0.021 ms. The symmetric encryption/decryption operation T_{A} is 0.02 ms. XOR, multiplication, and arithmetic operations have been neglected. The results are presented in Table 2, where T_{MTCD} and T_{gNB} represent the computation time for the MTCD and the gNB, respectively. We have only accounted for the operations during the first MTCD handover authentication and the group handover authentication of the BSPGH scheme.

Table 2. Computational costs of different protocols.

Protocol	T_{MTCD}	T_{gNB}	T_{MTCD}(μs)	T_{gNB}(μs)
5G-AKA	$4nT_H$	$2nT_H$	0.084n	0.042n
PPHAP	$3n(T_A + T_H)$-$2T_H$	$(2T_H+3T_A)$ n-T_H	0.123n − 0.042	0.102n − 0.021
NASS	nT_{RV}	$nT_H+2T_E + T_{RV}$	1.445n	0.021n + 2.775
BSPGH	$n(4T_H+2T_{PM})$	$(3n+3)T_H+(3n+5)T_{PM}+3nT_{PA}$	0.486n	0.669n + 1.068

In Table 2, "n" represents the number of MTCD to substitute specific numerical values. For the 5G-AKA scheme, the computation costs are $0.126n$ ms. For the PPHAP scheme, the computation costs are $0.205n$-0.063 ms. For the NASS scheme, the computation costs are $1.466n + 2.775$ ms. For the BSPGH scheme, the computation costs are $1.155n + 1.068$ ms. The NASS scheme involves complete RSA verification for handover authentication, which increases the overhead. The PPHAP scheme primarily uses symmetric encryption and hash operations for authentication, resulting in lower computational overhead but posing a risk of DDoS attacks. The 5G-AKA scheme has the lowest computational overhead but has serious security vulnerabilities. In contrast, the proposed BSPGH scheme is the most secure in terms of security functionality and has lower overhead compared to the NASS scheme. Overall, the BSPGH scheme has been proven to be the most effective.

7.3. Communication Cost

The communication costs for n MTCDs to perform handover authentication by the BSPGH scheme are evaluated and compared with the other three protocols. The communication costs include propagation time and transmission time. The propagation time is determined by the distance between the transmitter and the receiver and the propagation speed over the wireless communication channel, which is approximately 3×10^8 m/s. It is assumed that the radius of a cell is 200 m, and the data packets sent by an MTCD would take 200 m to propagate to the gNB at a speed of 3×10^8 m/s. According to the 3GPP standard on 5G communication, the downlink data rate for the urban general area scenario is 50 Mbps, and the uplink data rate is 25 Mbps [29]. In Table 3, "Amount of Information" refers to the total amount of data transmitted during the communication process. T_t and T_p represent transmission delay and propagation delay, respectively. "Up" and "Down" denote uplink and downlink data transmission. Therefore, we compare them separately. When calculating communication costs, we consider only the authentication phases during the handover, including the first MTCD handover authentication, group handover authentication.

Table 3. Communication costs of different protocols.

Protocol	Link	Amount of Information (bits)	T_t (μs)	T_p (μs)
5G-AKA	Up	128n	5.12n	0.67n
	Down	768n	15.36n	0.67n
PPHAP	Up	608n − 160	24.32n − 6.4	0.34n + 1.068
	Down	800n + 448	16n + 8.96	1.01n + 0.34
NASS	Up	512n + 288	20.48n + 11.52	0.67n
	Down	832n + 832	16.64n + 16.64	0.67n
BSPGH	Up	768n + 64	30.72n + 2.56	0.67(n + 1)
	Down	128n + 1824	2.56n + 36.48	0.67(n + 1)

Figure 7 demonstrates that when there is a large number of group MTCDs, the proposed BSPGH scheme exhibits superior performance in terms of communication overhead compared to the PPHAP and the NASS scheme. This advantage is attributed to the reduced information exchanged between MTCDs and gNBs in our protocol, leading to enhanced

efficiency. In contrast, the communication overhead of the 5G-AKA scheme is lower, but it suffers from a lot of security vulnerabilities. Overall, the results indicate the better performance achieved by the BSPGH scheme.

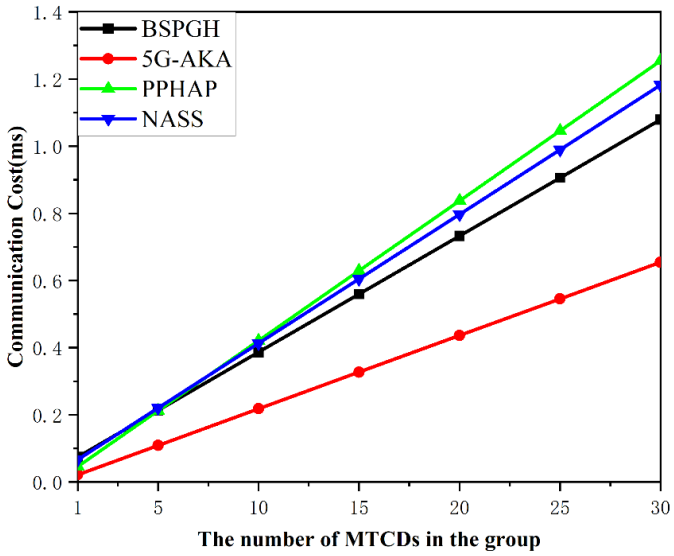

Figure 7. Comparison of communication cost.

7.4. Authentication Cost

Figure 8 shows the total time cost of the handover authentications for n MTCDs within a group. The comparison of four handover authentication schemes is as follows. The NASS scheme needs a far higher handover authentication cost compared to the BSPGH scheme, due to the involvement of complete RSA signature verification and modular exponentiation for computation and a higher communication cost during the authentication process; thus, it increases the overall latency. The PPHAP scheme has a lower handover authentication time cost, while the BSPGH scheme provides enhanced security features in mitigating DDoS attacks. The 5G-AKA scheme has a lower latency but suffers from serious security vulnerabilities. Therefore, although the BSPGH protocol inevitably introduces a minor computational overhead, it remains the most effective protocol in balancing the security functionality and system performance.

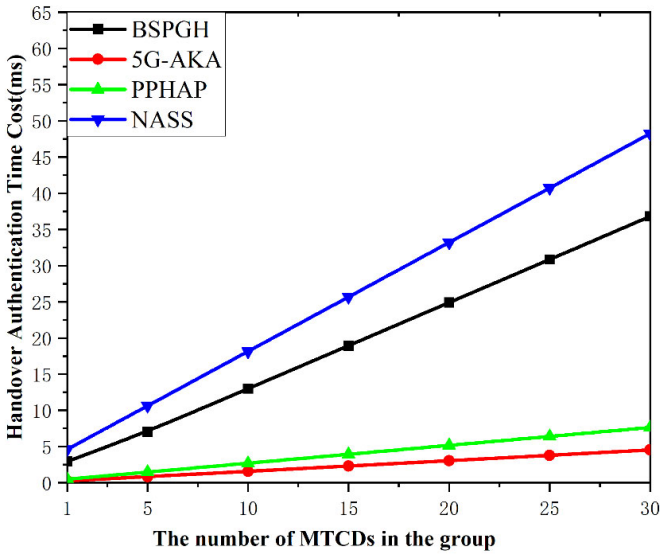

Figure 8. Comparison of the authentication cost.

We assess the robustness of the protocol for estimating the robustness of handover authentication. In the system, handover authentication may be forced to stop and restart when facing unknown attacks. We assume that unknown attacks can occur at each step of the handover authentication, and the probability of unknown attacks is uniform [1]. The average time of successful handover authentication is calculated as follows:

$$T = \frac{T_{success} + T_{failed}}{N_{success}} = \frac{\sum_{i=1}^{n} \frac{1}{n} \times t_{fail} \times p + t_{success} \times (1-p)}{1-p} \quad (38)$$

where T, $T_{success}$, and T_{failed} are the average time taken for a successful handover authentication, the total time of successful handover authentications, and the total time for failed handover authentications, respectively, $N_{success}$ is the number of successful handover authentications, p is the percentage of unknown attacks, n is the number of steps in the protocol, t_{fail} represents the total time cost before the attack happens in the i-th step, and t_{success} represents the time elapsed for one successful handover authentication before an attack occurs.

The simulation results of comparison with a group size of 30 MTCDs are shown as in Figure 9, when the percentage of unknown attacks increases. It is evident that the BSPGH protocol has a lower total time cost compared to NASS, indicating a better performance. However, it has a higher time cost compared to the solutions of the 5G standard and the PPHAP scheme. This is because that the 5G standard and the PPHAP scheme use a symmetric cryptographic system and have suffered security vulnerabilities. As shown in Figure 6, our proposed protocol can resist most major malicious attacks, albeit with a slightly higher total time cost. Overall, the proposed BSPGH scheme demonstrates a better performance in terms of security functionality and system efficiency.

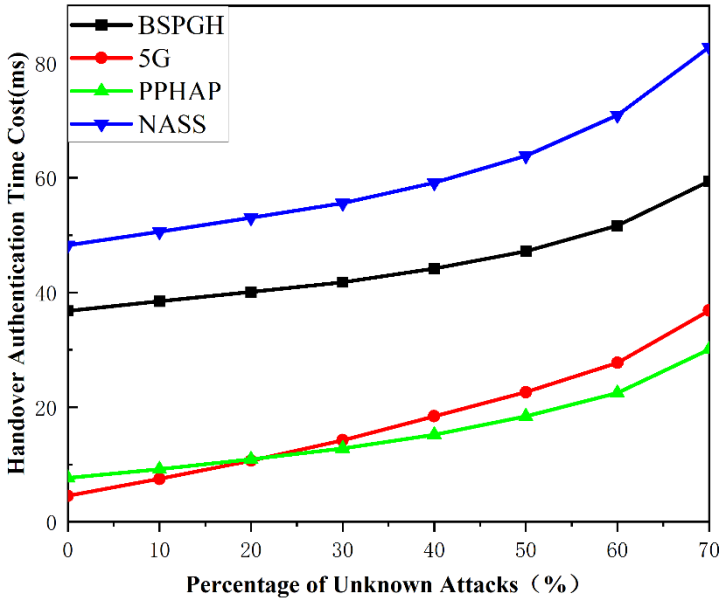

Figure 9. Comparison of the authentication cost for 30 MTCDs with unknown attacks.

7.5. Energy Consumption

The security protocols that offer the highest level of security functionality while consuming the least amount of energy are always the preferred choice due to the limited battery life of mobile devices like MTCDs. To evaluate the energy consumption of MTCDs, two factors should be considered including the energy required for data transmission and the energy consumed for the execution of the cryptographic functions. For the energy used for data transmission, the evaluation of energy consumption follows the data transmission

power model described in [28]. The calculation of the transmission energy cost for the uplink and downlink is as follows:

$$E_{ul} = (\alpha_u t_{udr} + \beta) \cdot t_{ul} \tag{39}$$

$$E_{dl} = (\alpha_d t_{ddr} + \beta) \cdot t_{dl} \tag{40}$$

where $\alpha_u = 438.39$ mW/Mbps, $\alpha_d = 51.97$ mW/Mbps, $\beta = 1288.04$ mW, $t_{udr} = 25$ Mbps, and $t_{ddr} = 50$ Mbps. t_{udr} represents the uplink throughput, t_{ddr} represents the downlink throughput, t_{ul} is the transmission time for uplink, and t_{dl} is the transmission time for downlink.

For the energy used for performing cryptographic functions, the way of approximation of energy cost in [30] is adopted. All experiments were conducted using a battery-powered Compaq iPAQ H3670 PDA, which is equipped with an Intel SA-1110 StrongARM processor running at 206 MHz, 64 MB RAM, and 16 MB FlashROM for evaluation. The energy cost of a single AES encryption or decryption operation is $7.87 + 1.21b$ µJ, where b is the number of bytes in the plaintext. The energy cost per byte for a SHA-1 hash operation E_H is 0.76 µJ. Considering that the energy cost for generating an ECDH public key is 276.7 mJ and for deriving an ECDH public key is 163.5 mJ, the energy consumption for a single scalar multiplication operation E_{PM} is approximately estimated to be 220.1 mJ, and the energy cost for a single RSA operation E_{RV} is 832.6 mJ. The energy consumption of using cryptographic primitives is derived based on the cryptographic operations used in the protocol, leading to a theoretical energy cost.

Figure 10 illustrates the relationship between the energy consumption during handover authentication and the number of MTCDs in a group. It is clearly visible from the figure that, compared to the NASS protocol, the energy consumption of BSPGH is lower and more efficient as the number of MTCDs increases. Additionally, the energy consumption of the 5G and PPHAP protocols is lower than our protocol. However, the 5G standard protocol has security issues and is susceptible to different malicious attacks, while the PPHAP protocol is also vulnerable to security threats like DDoS attacks. On the other hand, the BSPGH protocol can significantly enhance network security, albeit at a slightly higher cost compared to other protocols. It can also facilitate group handover authentication for a large number of users simultaneously. Therefore, the BSPGH protocol exhibits a better performance in terms of both security and efficiency.

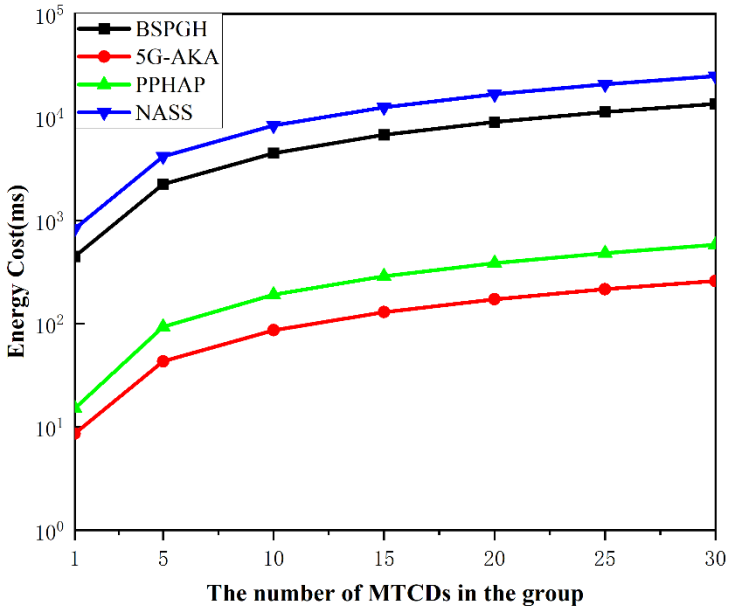

Figure 10. Comparison of the energy consumption.

7.6. Discussion of the Simulation Results

We have Table 4 to discuss the security features of the BSPGH, NASS, PPHAP, and 5G-AKA schemes in detail. The 5G-AKA scheme faces security problems in handover authentication including lack of forward secrecy for keys, vulnerability to DoS and DDoS attacks, session key leakage, susceptibility to linkability attacks, and the absence of mutual authentication between an MTC device and its target gNB. The NASS scheme is vulnerable to DoS and DdoS attacks. The PPHAP scheme struggles to resist DdoS attacks. On the other hand, the BSPGH scheme can overcome all of the abovementioned security vulnerabilities, ensuring the highest level of security during the handover authentication process.

Table 4. Security analysis of protocols.

Protocol	Security Features									
	MA	**PP**	**PFS**	**RA**	**IA**	**DoS**	**DDoS**	**SKL**	**SA**	**MITM**
BSPGH	√	√	√	√	√	√	√	√	√	√
5G-AKA		√		√	√				√	√
PPHAP	√	√	√	√	√	√		√	√	√
NASS	√	√	√	√	√			√	√	√

MA = mutual authentication; PP = privacy protection; PFS = perfect forward secrecy; RA = replay attack prevention; IA = impersonation attack prevention; DoS = DoS attack prevention; DDoS = DDoS attack prevention; SKL = session key leakage; SA = sybil attack prevention; MITM = man-in-the-middle attack prevention.

We have conducted experiments for performance comparison among the four handover authentication schemes in terms of handover authentication time for a group of 30 MTC devices. Our proposed protocol, the BSPGH scheme, is faster than the NASS scheme. This time improvement comes from the fact that the NASS scheme performs a full RSA signature verification and modular exponentiation calculations, which result in significant computational overhead, while the BSPGH approach works based on certificateless aggregate signatures and ECDH for session key generation, which need only point multiplication, point addition, and hashing operations so that relatively lower computational costs are incurred. The reduction in computational functions leads to a shorter authentication time, making the BSPGH scheme faster than the NASS scheme for handover authentication.

However, the BSPGH scheme takes a longer time for handover authentication compared to the 5G-AKA and the PPHAP schemes. This is because the BSPGH scheme employs asymmetric ECDH to replace the less secure symmetric key-based key derivation function (KDF) used in the 5G-AKA and the PPHAP scheme. The computational overhead of the ECDH is much higher than that of the KDF, but the KDF is weak to resistant exhaustive attacks, side-channel attacks, and replay attacks. By incorporating the ECDH, the BSPGH scheme cannot only address the security issues associated with key derivation functions but also ensure perfect forward secrecy/backward secrecy and the prevention of linkability attacks for session keys.

Considering the exponential growth in the number of 5G mobile devices in the future, the risks associated with DoS and DDoS attacks in large-scale MTCD scenarios will become more pronounced. Our solution effectively leverages the decentralized nature of blockchain, ensuring the authenticity and security of critical information, and more efficiently defending against DoS and DDoS attacks during the multi-user authentication process. Moreover, our protocol exhibits outstanding performance in the environments susceptible to these security attacks. Therefore, we believe that these overheads are tolerable to protect future 5G wireless networks.

8. Conclusions

In this paper, we have proposed the blockchain-assisted group handover authentication protocol for MTC communication in 5G wireless networks. The BSPGH protocol

aims to reduce the authentication time when multiple MTCDs undergo frequent handovers between gNBs by adopting group handover authentication. The proposed BSPGH scheme requires no modification to the existing 5G network architecture specified by the 3GPP standard, making it easy to deploy. By formal verification using Scyther Tool and BAN-logic, we have analyzed the protocol's security properties, demonstrating that it can provide perfect forward secrecy and resist impersonation attacks, DoS/DDoS attacks, and some other attacks. It also ensures the anonymity of the MTCDs. Moreover, the performance analysis has shown that the BSPGH scheme can meet the requirements of various application scenarios to confirm its efficiency.

Author Contributions: Conceptualization, R.M., J.Z. and M.M.; methodology, R.M.; software, R.M.; validation, J.Z. and M.M.; formal analysis, R.M.; investigation, R.M.; resources, J.Z. and M.M.; data curation, R.M.; writing—original draft preparation, R.M.; writing—review and editing, J.Z. and M.M.; visualization, R.M.; supervision, J.Z. and M.M.; project administration, J.Z. and M.M.; funding acquisition, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: The work of this paper was funded by the National Natural Science Foundation of China (No. 62171387), and the China Postdoctoral Science Foundation (No. 2019M663475).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yan, X.; Ma, M. A privacy-preserving handover authentication protocol for a group of MTC devices in 5G networks. *Comput. Secur.* **2022**, *116*, 102601. [CrossRef]
2. Shariatmadari, H.; Ratasuk, R.; Iraji, S.; Laya, A.; Taleb, T.; Jäntti, R.; Ghosh, A. Machine-type communications: Current status and future perspectives toward 5G systems. *IEEE Commun. Mag.* **2015**, *53*, 10–17. [CrossRef]
3. Ahmad, I.; Shahabuddin, S.; Kumar, T.; Okwuibe, J.; Gurtov, A.; Ylianttila, M. Security for 5G and beyond. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3682–3722. [CrossRef]
4. Sharma, A.; Jain, A.; Sharma, I. Exposing the security weaknesses of fifth generation handover communication. In Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, 6–8 July 2019; pp. 1–6.
5. Gupta, S.; Parne, B.L.; Chaudhari, N.S. SRGH: A secure and robust group-based handover AKA protocol for MTC in LTE-A networks. *Int. J. Commun. Syst.* **2019**, *32*, e3934. [CrossRef]
6. Basudan, S. LEGA: A lightweight and efficient group authentication protocol for massive machine type communication in 5G networks. *J. Commun. Inf. Netw.* **2020**, *5*, 457–466. [CrossRef]
7. Lai, C.; Ma, Y.; Lu, R.; Zhang, Y.; Zheng, D. A novel authentication scheme supporting multiple user access for 5G and beyond. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 2970–2987. [CrossRef]
8. Aydin, Y.; Kurt, G.K.; Ozdemir, E.; Yanikomeroglu, H. A flexible and lightweight group authentication scheme. *IEEE Internet Things J.* **2020**, *7*, 10277–10287. [CrossRef]
9. Dwivedi, S.K.; Amin, R.; Vollala, S.; Khan, M.K. B-HAS: Blockchain-Assisted Efficient Handover Authentication and Secure Communication Protocol in VANETs. *IEEE Trans. Netw. Sci. Eng.* **2023**, *10*, 3491–3504. [CrossRef]
10. Soni, M.; Singh, D.K. Blockchain-based group authentication scheme for 6G communication network. *Phys. Commun.* **2023**, *57*, 102005. [CrossRef]
11. Cai, J.; Tao, X.; Wang, C. Cooperative Authentication Scheme for Heterogeneous Networks Based on Identity Group Signature and Blockchain. *IEEE Trans. Veh. Technol.* **2023**, *73*, 1394–1399. [CrossRef]
12. Son, S.; Lee, J.; Park, Y.; Park, Y.; Das, A.K. Design of blockchain-based lightweight V2I handover authentication protocol for VANET. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 1346–1358. [CrossRef]
13. Shawky, M.A.; Jabbar, A.; Usman, M.; Imran, M.; Abbasi, Q.H.; Ansari, S.; Taha, A. Efficient blockchain-based group key distribution for secure authentication in VANETs. *IEEE Netw. Lett.* **2023**, *5*, 64–68. [CrossRef]
14. Ma, R.; Cao, J.; Feng, D.; Li, H.; He, S. FTGPHA: Fixed-trajectory group pre-handover authentication mechanism for mobile relays in 5G high-speed rail networks. *IEEE Trans. Veh. Technol.* **2019**, *69*, 2126–2140. [CrossRef]
15. Yang, Y.; Cao, J.; Ma, R.; Cheng, L.; Chen, L.; Niu, B.; Li, H. FHAP: Fast Handover Authentication Protocol for High-Speed Mobile Terminals in 5G Satellite-Terrestrial Integrated Networks. *IEEE Internet Things J.* **2023**, *10*, 13956–13973. [CrossRef]

16. Aydin, Y.; Kurt, G.K.; Ozdemir, E.; Yanikomeroglu, H. Group handover for drone base stations. *IEEE Internet Things J.* **2021**, *8*, 13876–13887. [CrossRef]
17. Alnashwan, R.; Gope, P.; Dowling, B. Privacy-aware secure region-based handover for small cell networks in 5G-enabled mobile communication. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 1898–1913. [CrossRef]
18. Li, X.; Yin, X.; Ning, J. RelCLAS: A Reliable Malicious KGC-Resistant Certificateless Aggregate Signature Protocol for Vehicular Ad Hoc Networks. *IEEE Internet Things J.* **2023**, *10*, 21100–21114. [CrossRef]
19. Chaer, A.; Salah, K.; Lima, C.; Ray, P.P.; Sheltami, T. Blockchain for 5G: Opportunities and challenges. In Proceedings of the 2019 IEEE Globecom Workshops (GC Wkshps), Waikoloa, HI, USA, 9–13 December 2019; pp. 1–6.
20. Secp256k1: A Key Algorithm in Cryptocurrencies. August 2023. Available online: https://www.nervos.org/knowledge-base/secp256k1_a_key%20algorithm_(explainCKBot) (accessed on 30 March 2024).
21. 3GPP TS 23.501 Release 16. 2020. Available online: https://www.3gpp.org/specifications-technologies/releases/release-16 (accessed on 30 March 2024).
22. Ge, X.; Tu, S.; Mao, G.; Wang, C.-X.; Han, T. 5G ultra-dense cellular networks. *IEEE Wirel. Commun.* **2016**, *23*, 72–79. [CrossRef]
23. Dolev, D.; Yao, A. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–208. [CrossRef]
24. Lee, H.; Kim, D.; Chung, B.; Yoon, H. Adaptive hysteresis using mobility correlation for fast handover. *IEEE Commun. Lett.* **2008**, *12*, 152–154.
25. Burrows, M.; Abadi, M.; Needham, R. A logic of authentication. *ACM Trans. Comput. Syst. (TOCS)* **1990**, *8*, 18–36. [CrossRef]
26. Cremers, C.J.F. Scyther: Semantics and Verification of Security Protocols. 2006. Available online: https://www.semanticscholar.org/paper/Scyther-:-semantics-and-verification-of-security-Cremers/4abeecda63a90f2c39554843c8189a6cd8c4eea5 (accessed on 30 March 2024).
27. Chow, M.C.; Ma, M. A secure blockchain-based authentication and key agreement scheme for 3GPP 5G networks. *Sensors* **2022**, *22*, 4525. [CrossRef] [PubMed]
28. Barker, E. *Recommendation for Key Management Part 1: General*; NIST Special Publication 800-57 Part 1 Revision 4; National Institute of Standards and Technology (NIST): Gaithersburg, MD, USA, 2016; Volume 57, pp. 1–142.
29. 3GPP TS 22.261 Release 16. April 2021. Available online: https://www.etsi.org/deliver/etsi_ts/122200_122299/122261/16.14.00_60/ts_122261v161400p.pdf (accessed on 30 March 2024).
30. Potlapally, N.R.; Ravi, S.; Raghunathan, A.; Jha, N.K. A study of the energy consumption characteristics of cryptographic algorithms and security protocols. *IEEE Trans. Mob. Comput.* **2005**, *5*, 128–143. [CrossRef]