Future-Proofing Authentication Against Insecure Bootstrapping for 5G Networks: Feasibility, Resiliency, and Accountability

Saleh Darzi[a], Mirza Masfiqur Rahman[b], Imtiaz Karim[c], Rouzbeh Behnia[d], Attila Altay Yavuz[a] and Elisa Bertino[b]

[a]Bellini College of Artificial Intelligence, Cybersecurity and Computing, University of South Florida, Tampa, 33620, Florida, USA
[b]Department of Computer Science, Purdue University, West Lafayette, 47907, Indiana, USA
[c]Department of Computer Science, University of Texas at Dallas, Richardson, 75080, Texas, USA
[d]School of Information Systems at University of South Florida, Tampa, 33620, Florida, USA

ARTICLE INFO

Keywords:
5G Cellular Networks
Authentication
Network Performance Analysis
Transitional Post-Quantum Security

ABSTRACT

The 5G protocol lacks a robust base station (BS) authentication mechanism during the initial bootstrapping phase, leaving it susceptible to fake BSs, spoofed broadcasts, and large-scale manipulation of System Information Blocks (SIBs). Existing solutions incur high communication overhead, rely on centralized trust, and lack accountability and long-term breach resiliency. Given the inevitability of BS compromise and the severe impact of forged SIBs as the root of trust (e.g., fake alerts, tracking, false roaming), distributed trust, verifiable forgery detection, and audit logging are essential yet remain largely unexplored. These challenges are further amplified by the emergence of quantum-capable adversaries. While NIST Post-Quantum Cryptography (PQC) standards are widely viewed as a path toward long-term security, their feasibility under 5G's strict packet-size, latency, and broadcast constraints has not been systematically studied. This work presents, to our knowledge, the first comprehensive network-level performance characterization of integrating NIST-PQC standards and conventional digital signatures into 5G BS authentication, showing that direct PQC adoption is impractical due to excessive signature sizes, fragmentation, and protocol-level delays. To address these challenges, we propose BORG, a future-proof authentication framework based on a Hierarchical Identity-Based Threshold Signature with Fail-Stop (HITFS) properties. BORG distributes trust across multiple BSs via threshold signing, enables post-mortem verifiable forgery detection, and provides tamper-evident, PQ-secure audit logging, while maintaining compact signatures that fit within a single SIB1 packet without fragmentation and incurring minimal UE overhead, as validated through our real over-the-air 5G testbed implementation.

1. Introduction

Despite advancements in next-generation cellular networks, the absence of a secure and efficient bootstrapping mechanism between User Equipments (UEs) and Base Stations (BSs) critically undermines the security of the 5G networks. The bootstrapping protocol enables UEs to connect to BSs and access the core network. However, during the initial Radio Resource Control (RRC) connection, the UE selects a BS based solely on signal strength and broadcast parameters [1], without any verified BS identity and authentication. This absence of a robust or standardized BS authentication leads to severe security attacks, such as fake base stations, phishing, and spoofed emergency alerts [2, 3].

1.1. Prior Works on BS Authentication

To address the absence of BS authentication, prior efforts have pursued several broad directions. Certificate-based approaches adapt Public Key Infrastructure (PKI) frameworks to attach digital signatures and certificate chains to SIB messages, providing verifiable BS identity at the cost of significant communication overhead [4, 5, 6, 7]. To reduce this overhead, certificate-free designs based on Identity-Based Signatures (IBS) derive BS and AMF keys hierarchically from a master key pre-installed in the USIM, eliminating certificate transmission and achieving more compact footprints [8, 9]. Token-based and symmetric schemes offer lightweight pre-authentication without asymmetric overhead but do not protect SIB content itself [10, 11]. More recent efforts have explored threshold signatures to distribute signing responsibility across multiple BSs [12, 13], and hybrid constructions have begun addressing long-term security for cellular authentication [14, 15, 16].

Building on these solutions, various optimizations, such as efficient certificate delivery [17, 18], online-offline signatures [12], and outsourced computation via auxiliary entities [19], have been proposed to reduce signing and certificate overheads [2], with recent works further advancing efficiency through certificate-free designs based on hierarchical IBSs [20]. Although recent efforts from 3GPP consider protecting the unicast RRC messages, initial bootstrapping messages like SIB1 are still unprotected. In particular, TS 38.331 Annex B.1 states that even after Access Stratum (AS) security activation, SIB1 can be sent without integrity protection and ciphering [21]. Below, we outline critical research gaps and the security requirements for 5G BS authentication.

salehdarzi@usf.edu (S. Darzi); rahman75@purdue.edu (M.M. Rahman); imtiaz.karim@utdallas.edu (I. Karim); behnia@usf.edu (R. Behnia); attilaayavuz@usf.edu (A.A. Yavuz); bertino@purdue.edu (E. Bertino)
ORCID(s):

(i) Lack of Efficient Distributed Authentication in 5G: The security of 5G bootstrapping currently depends on a single BS. However, modern cellular deployments increasingly rely on heterogeneous, densely deployed, and physically exposed infrastructures (e.g., small cells, femtocells, and O-RAN units [22, 23]), where software modification and physical access significantly expand the attack surface [24]. Thus, individual BSs are more susceptible to compromise through rooting, tampering, or other software exploits [25]. This creates a structural single point of failure: compromise of even one BS can break authentication guarantees for all UEs under its coverage, enabling fake-BS attacks, SIB manipulation, tracking, and DoS. The KT femtocell attack [26] demonstrates this risk in practice, showing that inexpensive, user-provisioned hardware can be rooted and used to inject malicious SIB1 or control uplink/downlink communication, making single-BS trust inadequate.

Furthermore, multi-connectivity, where UEs interact with Master and Secondary gNBs (EN-DC, NE-DC), carrier-aggregation cells, CoMP clusters, and Xn-coordinated nodes, is already the operational norm in 5G [2]. Leveraging multiple BSs for authentication therefore aligns with existing architectural practices and removes the fundamental mismatch between multi-BS communication and single-BS trust. Even when a single logical BS serves a region, modern 5G RAN architectures increasingly virtualize and disaggregate BS functions using software-defined and cloud-native principles, allowing protocol and security operations to be executed across multiple virtualized components and compute nodes rather than a single physical entity. Thus, an effective authentication solution must distribute trust across multiple BSs or instances thereof. Threshold signatures (e.g., [27]) enable this by requiring a quorum of independent stations to collaboratively authenticate broadcast system information, thereby eliminating the single point of failure, reducing BS/key compromise risks [28], strengthening attacker resistance, and aligning with existing 5G architectural patterns. Despite this need, only a few efforts [13, 12] have explored threshold signatures for 5G authentication. Moreover, UEs resource constraints, limited packet sizes, and frequent broadcasts, especially in distributed BS settings, make efficiency crucial. Thus, the designed authentication must minimize signature, communication, and storage overhead while ensuring fast signing and verification.

(ii) Lack of Accountability, Breach Resiliency, and Long-Term Security in 5G Authentication: Given that the SIB message serves as the root of trust in 5G bootstrapping, any forged signature effectively gives an attacker control over a UE's view of the network and enables impactful attacks such as large-scale impersonation, MITM, fake alerts, tracking, stealthy DoS [29], and false roaming. Because BS compromises are ultimately inevitable—whether through physical access, software exploits, advanced processing power, or insider threats—prevention alone is insufficient. A robust authentication design must support reliable detectability and distinguish malicious forgeries from benign failures rather than relying solely on idealized security assumptions.

This need becomes even more critical when viewed alongside real-world intrusion behavior: industry evidence shows that security breaches often persist undetected for long periods. Data breach reports (e.g., IBM [30], Verizon [31]) estimate an average dwell time of roughly 204 days before detection, giving attackers months to escalate privileges, manipulate system behavior, and stage follow-on operations without being noticed. In the case of 5G, the recent SK Telecom data breach shows the attackers were in the system and waited to affect 28 servers before doing the data breach [32]. Similarly, a forged SIB or compromised BS could remain active for extended periods, silently influencing mobility, routing, and service selection. Once an adversary injects a malicious broadcast, it can repeatedly mislead UEs or force attachment to rogue cells, enabling sustained exploitation. Without a tamper-evident mechanism to detect such manipulation, these attacks can remain invisible indefinitely.

Moreover, in security-critical infrastructures, such as 5G, the ability to prove that a breach occurred is as important as preventing one. Yet current 5G architecture provides no cryptographically enforced provenance of broadcast signatures, no mechanism to identify compromised signers, and no way to produce irrefutable evidence of misbehavior. This absence prevents operators from halting further exploitation, guiding remediation, or performing forensics after an attack. Verifiable forgery detection and tamper-evident logging are therefore essential for post-mortem analysis, accountability, and long-term integrity, particularly when logs from multiple BSs or core entities can be cross-validated to detect inconsistencies, forks, or replay attempts. However, existing RAN logs are purely operational, lack cryptographic binding, and are susceptible to modification by compromised software or attackers with elevated privileges, leaving no trustworthy link between broadcast signatures and recorded events. Consequently, 5G authentication faces a pressing challenge: the absence of a distributed and verifiable audit logging mechanism that provides long-term security and non-repudiation.

(iii) Lack of Systematic Feasibility Analysis for Long-Term Security: These challenges are further amplified once large-scale quantum computers emerge, as quantum algorithms are expected to undermine the computational hardness assumption of the classical public-key primitives. Combined with the inevitability of BS compromise and practical implementation failures (e.g., side-channel attacks), this means that signature forgeries will occur over time, reinforcing the need to future-proof 5G authentication. This urgency is also reflected in global efforts initiated by the National Institute of Standards and Technology (NIST) on standardizing Post-Quantum Cryptography (PQC) [33] and the migration guidelines of the European Telecommunications Standards Institute (ETSI) and IEEE Standards Association, which emphasize preparing communication systems for the Post Quantum (PQ) era [34]. While preliminary integration of NIST-PQC algorithms into various network protocols, such as TLS [35] and PQ-WireGuard,

has begun, these efforts reveal significant trade-offs, highlighting the substantial overhead and limited practicality of current PQ schemes in constrained, latency-sensitive mobile environments. For 5G BS authentication, these limitations are even more restrictive due to strict size, timing, and broadcast constraints; for example, initial frame synchronization messages like SIB1 are limited to 372 bytes and are transmitted periodically with a delay around 160 ms (discussed in detail in Section 6). Yet the feasibility of integrating NIST-PQC signatures into 5G bootstrapping, particularly in distributed or hierarchical settings with protocol-level intricacies, has not been comprehensively analyzed, leaving a critical open question in the evolution of secure 5G authentication.

1.2. Our Contributions

To address these challenges, we begin by evaluating the feasibility of integrating NIST-PQC standards into 5G BS authentication through a protocol-level performance analysis. Our findings highlight severe performance bottlenecks that arise when directly applying NIST-PQC signatures to 5G BS authentication. Motivated by these limitations, we present BORG, a future-proof authentication framework that provides efficient threshold (distributed) signing with post-mortem (PM) forgery detection and a verifiable auditing mechanism. Our key contributions are as follows:

(i) Analysis of NIST-PQC Standards and Conventional Alternatives: To our knowledge, this is the first in-depth evaluation of the NIST-PQC scheme adoption in the context of 5G BS authentication. Focusing on SIB1 as the critical broadcast message in the bootstrapping process, we show that directly applying NIST-PQC signatures is impractical, as detailed in Section 4. Specifically, the large signature and certificate sizes of NIST's primary (lattice-based) PQC standard *ML-DSA* [36] impose a 12276-byte communication overhead, requiring fragmentation and causing significant 5G packet delays. This results in a total end-to-end delay of 5282 ms, which is incompatible with the real-time requirements of 5G BS communication. We also assess conventional hierarchical IBS schemes to broaden the performance profile; while offering good performance [8], they lack support for distributed trust, accountability, and long-term forgery detection.

(ii) BORG: An Efficient, Distributed, and Accountable Authentication Framework: Given the infeasibility of NIST-PQC signatures for 5G BS authentication, we design BORG, a future-proof authentication framework based on a Hierarchical Identity-Based Threshold Signature with Fail-Stop (HITFS) property. *(1)* BORG distributes trust across multiple BSs via threshold signatures, ensuring resiliency even when some BSs are compromised or misbehave. *(2)* It provides conventionally secure authentication with post-mortem, verifiable forgery detection through a fail-stop mechanism. Even if the underlying classical hardness assumptions (e.g., discrete logarithms) fail in the future, honest and computationally bounded signers can identify and prove forgeries, including those produced by a quantum-capable adversary, thereby ensuring strong accountability and long-term security. This forgery-detection capability is reinforced through a distributed audit logging mechanism, ensured via PQ-secure threshold signatures, ensuring tamper-evident records, non-repudiation, and long-term system resiliency. *(3)* Despite supporting distributed trust and accountability, BORG maintains compact signatures, low communication overhead, minimal UE computation, and reduced end-to-end delay, eliminating the need for fragmentation. Compared to the existing conventional-secure alternatives like *Schnorr-HIBS* (see TABLE 1 & 2), BORG achieves similar overhead while additionally offering distributed authentication, accountability, and forgery detection.

(iii) Open-Sourced Evaluation Framework: We fully implemented BORG by incorporating it into a real 5G testbed in srsRAN and then conducted an extensive performance evaluation against existing authentication schemes. Tested with over-the-air 5G communication, BORG demonstrates practical deployability with low computational and communication overhead. Compared to aggregate signature schemes such as *BLS*, BORG achieves faster signing and lower end-to-end delay with reduced communication overhead in 5G settings. Relative to IBS schemes [8], BORG attains similar runtime while also providing distributed authentication, audit logging, and post-mortem forgery detection. Moreover, BORG is up to three orders of magnitude faster and incurs 85× less communication overhead than NIST-PQC's *ML-DSA* [36]. These results highlight BORG as a compact, efficient, and future-proof solution for 5G bootstrapping authentication. To support reproducibility, we publicly release the complete source code of BORG[1], along with its over-the-air 5G testbed implementation[2].

1.3. Outline

The remainder of this paper is organized as follows. Section 2 presents the system model, notation, and cryptographic building blocks underlying BORG. Section 3 formalizes the threat model, scope, and security definitions. Section 4 analyzes the feasibility of integrating NIST-PQC standards into 5G BS authentication, demonstrating the impracticality of direct adoption. Section 5 presents the BORG framework, including the proposed scheme, security analysis, and full 5G protocol instantiation. Section 6 provides a comprehensive performance evaluation against PQC and conventional baselines on a real over-the-air 5G testbed. Section 7 surveys related work. Section 8 concludes the paper and outlines future directions.

2. Preliminaries and Building Blocks

This section outlines the notation, network architecture, and the cryptographic building blocks.

Notations: The symbol || denotes concatenation, and · denotes multiplication. For two primes p and q, let $\mathbb{Z}_q$ be the finite field of integers modulo q, and let $\mathbb{G}$ be a cyclic

[1] github.com/TheSalehDarzi/BORG-Scheme/tree/main/BORG

[2] github.com/TheSalehDarzi/BORG-Scheme/tree/main/OTA

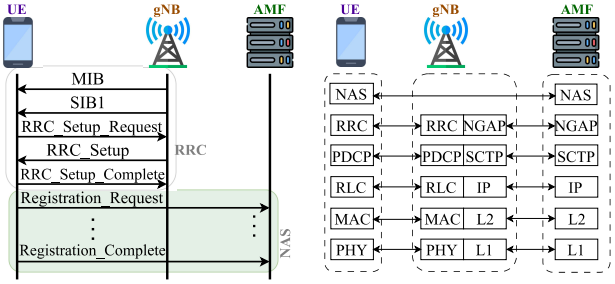


Figure 1: Initial 5G network connection setup and protocol stack.

group of prime order p with generator g. We define two cryptographically secure hash functions: $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q$ and $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_q$. $x \xleftarrow{\$} S$ indicates that x is sampled uniformly at random from the set S. Vectors are denoted by $\vec{x}$, and $\{x_i\}_{i=1}^{n} = \{x_1, x_2, \ldots, x_n\}$ represents a set of n elements. Finally, sk, PK, and ID refer to the secret key, public key, and the identity of an entity (e.g., MAC address), respectively. A list of acronyms is provided in Appendix A.1.

2.1. System Model: 5G Cellular Network

2.1.1. Network Components

The 5G network consists of three main entities [37]:

- ***5G Network Core:*** This entity serves as the central management of the cellular network, responsible for service delivery, session management, policy control, data handling, and security enforcement while integrating multiple network functions. One of the crucial components of the network core is the Access and Mobility Management Function (AMF), which is most relevant to our work.
- ***User Equipment (UE):*** Located at the network edge, a UE refers to a cellular device (e.g., smartphone or IoT) subscribed to the network. Each UE is registered and equipped with a Universal Subscriber Identity Module (USIM) issued by the network authorities. It uses a unique identifier for communication, connection establishment, and access to network services.
- ***Radio Access Network (RAN):*** This network, comprising BSs (gNB) and UEs, manages radio transmissions, traffic, data exchange, and user service requests. Our work focuses on this component, where bootstrapping and system information messages are periodically broadcast. As these messages are neither encrypted nor signed, they are susceptible to adversarial manipulation. The UE initiates service requests procedures based on the content of these broadcasts and the type of service required.

2.1.2. Initial BS-UE Communication

In 5G protocol stack (Figure 1), the topmost layer in the BS is Radio Resource Control (RRC). For UE and AMF, the Non-Access Stratum (NAS) layer is stacked over the RRC layer. Only the master public key (PK_{ID_0}) is securely embedded in the USIM and is assumed to be publicly verifiable. Private keys for AMF and BSs are derived from the master secret key (sk_{ID_0}) and distributed through secure channels. For initial bootstrapping, the BS broadcasts the System Information (SI)—an RRC message—to the UE, announcing its configuration parameters. SI consists of the Master Information Block (MIB) and the System Information Block (SIB). The MIB, a short message, assists in decoding the first SIB: SIB1. Broadcast over the Downlink Shared Channel (DL-SCH), SIB1 contains scheduling and availability information for other SIB messages. As per 3GPP specifications [21], SIB1 has a maximum size of 372 bytes and is transmitted periodically every 160 ms, with repeated broadcasts allowed. Fig. 2 illustrates the structure of SIB1. Some fields are always present, while others are conditional or optional. The *Cell Selection Info* field provides signal quality metrics, while *Cell Access Related Info* includes Public Land Mobile Network (PLMN) identifiers and cell access status. Optional fields like *IMS-Emergency Support* indicate support for emergency services in limited service mode. For detailed field descriptions, see [21]. After receiving SIB1, the UE initiates the RRC setup. Upon successful RRC connection, the UE initiates NAS registration with the AMF. Several additional SIB messages (SIB2–SIB21) are transmitted over DL-SCH in periodic windows, each serving specific functions. For instance, SIB3 provides NR *intra*-frequency neighbor cell lists and reselection areas, SIB4 conveys *inter*-frequency equivalents, SIB9 delivers GPS and UTC time, and SIB15 carries disaster roaming configurations.

2.2. Building Blocks

Hierarchical Identity-Based Threshold Signature Scheme with Fail-Stop property (HITFS). Our proposed BORG framework realizes a Hierarchical Identity-Based Threshold Signature scheme with Fail-Stop property (HITFS), which harnesses a Hierarchical IBS (HIBS)[38, 8] and FROST [27]. In the HIBS model, keys are derived hierarchically, where each level's keys are generated from its parent, binding identities directly to signing keys without the need for a trusted certificate authority. This structure supports efficient identity validation and key expiration verification using a compact master key [19, 38, 39]. By incorporating threshold cryptography, any t out of n authorized signers can collaboratively produce a valid signature without reconstructing the group's secret key, while fewer than t participants are cryptographically incapable of forging a signature. This design is particularly well-suited for distributed and fault-tolerant signing, as each participant only holds a share of the signing key. As long as the number of colluding parties remains below the threshold t, unauthorized signing remains infeasible [40]. In addition, BORG integrates an FS security mechanism [41, 42], which leverages the second pre-image resistance of cryptographic hash functions to enable post-mortem forgery detection against quantum adversaries. While FS operates similarly to standard signatures under conventional security assumptions, it uniquely allows a signer to prove that a forgery has occurred if those assumptions are violated. This serves as a breach resiliency

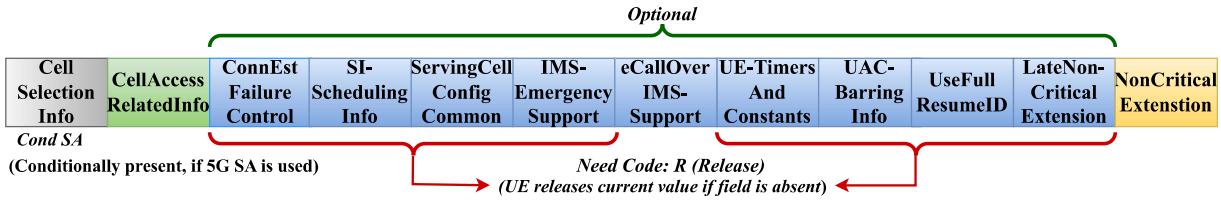


Figure 2: SIB1 message structure in 5G.

mechanism, halting further key usage and providing cryptographic evidence to absolve the signer of liability.

Definition 2.1. *A hierarchical identity-based threshold signature scheme with fail-stop property is a 7-tuple algorithm as shown below:*

- $(sk_{ID_0}, PK_{ID_0}, params) \leftarrow$ `HITFS.Setup`(1^κ)*: Given the security parameter* κ*, it outputs the master secret and public keys* (sk_{ID_0}, PK_{ID_0}) *and the system parameters, params, which is an implicit input to all the following algorithms.*
- $(\{sk_{ID_{k,i}}\}_{i=1}^n, \vec{Q}_{ID_k}) \leftarrow$ `HITFS.Extract`$(\vec{ID}_k, \vec{Q}_{ID_{(k-1)}}, sk_{ID_{(k-1)}})$*: Given the identity vector at level* k $\vec{ID}_k = (ID_1, ID_2, \ldots, ID_k)$*, the algorithm extracts the secret key of* ID_k *using the public key vector* $\vec{Q}_{ID_{(k-1)}}$ *and the secret key* $sk_{ID_{(k-1)}}$ *from level* $k-1$*. It then outputs the secret key shares for each participant* $(\{sk_{ID_{k,i}}\}_{i=1}^n)$ *and computes the corresponding group public key values* $\vec{Q}_{ID_k} = (Q_{ID_1}, Q_{ID_2}, \ldots, Q_{ID_k})$.
- $\mathcal{L}_i \leftarrow$ `HITFS.Preprocess`(J)*: Given the predetermined number of messages to be signed* J*, it returns the commitment values for all participants* $i \in [1, n]$ *in a list* $\mathcal{L}_i \leftarrow (i, \{E_{i,j}\}_{j=1}^J, \{D_{i,j}\}_{j=1}^J)$.
- $\sigma_{k,j} \leftarrow$ `HITFS.Sign`$(m_j, \mathcal{L}_i, \{sk_{k,i}\}_{i=1}^\beta)$*: Given a message* m_j *with index* j*, commitmetn values* $\mathcal{L}_i$*, and* $\beta \in [t, n]$ *participating signers' secret keys* $(\{sk_{k,i}\}_{i=1}^\beta)$*, it returns a signature* $\sigma_{k,j}$ *for signers at level* k.
- $\{0, 1\} \leftarrow$ `HITFS.MVerify`$(m_j, \vec{ID}_k, \vec{Q}_{ID_k}, \sigma_{k,j})$*: It returns* 1 *if the signature* $\sigma_{k,j}$ *on message* m_j *is valid with respect to the identity vector* $\vec{ID}_k$ *and public key vector* $\vec{Q}_{ID_k}$*, and* 0 *otherwise.*
- $\pi \leftarrow$ `HITFS.PoF`$(\{\hat{e}_{i,j}\}_{i=1}^\beta, \{\hat{d}_{i,j}\}_{i=1}^\beta, m, \sigma'_k, hist)$*: Given the message-signature pair* (m, σ'_k)*, random commitment values of the signing participants, and the history of previous signatures* $(hist)$*, it outputs* π*, a proof of forgery if* σ'_k *is forged; otherwise, it returns "Not A Forgery".*
- $\{0, 1\} \leftarrow$ `HITFS.PoFVerify`$(\alpha_k, sk_{k-1}, Q_{ID_k}, m, \sigma'_k, \pi)$*: On the selected random input* α_k*, public* Q_{ID_k}*, secret key* sk_{k-1}*, message* m*, signature* σ'_k*, and* π*, it returns* 1 *if the proof of forgery is valid, otherwise,* 0.

Definition 2.2. *Discrete Logarithm Problem (DLP) [43]: Let* $\mathbb{G}$ *be the finite cyclic group with generator* g*, given* $g \in \mathbb{G}$*,* $h \in \mathbb{G}$*, and* $h = g^x \mod p$ *with some unknown* $x \in \mathbb{Z}_q$*, the* $(EC)DLP$ *requires computing* $x = \log_g h \mod p$.

PQ Threshold Digital Signature Scheme. We employ a PQ-secure threshold signature scheme (`ThPQ`) for distributed audit logging, eliminating single points of failure. `ThPQ` is critical for audit logging and subsequent forgery detection in our future-proofed defense against quantum-capable adversaries. It distributes the audit key across multiple BSs, allowing any t of them to jointly sign, while preventing forgery by up to $(t-1)$ compromised nodes [44]. The scheme consists of: *(i)* `ThPQ.KeyGen`$(1^\kappa, t, n)$: generates a global public key PK and a set of n secret key shares $(sk_1, \ldots, sk_n)$ from the security parameter κ and threshold t out of n; *(ii)* `ThPQ.Sign`(sk_i, m): each signer i for $i = 1, \ldots, t$, uses sk_i to produce a signature share σ_i on the message m; *(iii)* $\sigma \leftarrow$ `ThPQ.Aggregate`$(\{\sigma_i\}_{i=1}^t)$ aggregates t signature shares into a valid signature σ. *(iv)* $\{0, 1\} \leftarrow$ `ThPQ.Verify`(PK, m, σ) verifies σ on message m using PK. For further details, see [44].

3. Threat and Security Models

This section outlines the threat model and the scope of our solution followed by the security model that underpin the formal security proof of `BORG`.

3.1. Threat Model and Scope

We consider a probabilistic polynomial-time (PPT) adversary with full control over the wireless medium. The adversary can eavesdrop on all broadcast messages, inject, modify, or replay forged SIB messages, and impersonate legitimate base stations (gNBs) to mislead UEs. Additionally, the adversary may corrupt up to $(t-1)$ BSs, gaining access to their secret keys and internal states to craft forgeries. The adversary is thus capable of performing three attack vectors commonly exploited in cellular networks, as captured in our threat model and illustrated in Fig. 3, and detailed below:

- **Fake Base Stations (FBSs).** These attacks [45] are carried out by luring the victim UE to connect to an FBS that spoofs legitimate BSs. Once connected, attackers can launch multi-phase attacks that exploit vulnerabilities in subsequent protocol stages [46].
- **Key Compromise Scenarios.** We account for active adversaries capable of compromising BSs to extract signing keys [28], forge signatures, and impersonate legitimate BSs during 5G bootstrapping [47]. While some BSs may be compromised, we assume at least t out of n remain uncompromised. This is a practical assumption, as a majority compromise would indicate that the entire network

is no longer trustworthy. To support this, BS hardening techniques such as advanced intrusion detection and secure configuration practices can be applied [48].

- **MiTM Attacker.** An MiTM attacker impersonates a BS to a victim UE and vice versa, enabling interception, modification, or replay of messages. This is possible when traffic is not protected (e.g., digitally signed) [24].

Our threat model also captures provable detection of quantum-capable adversaries with the potential to break conventional signatures [49]. While our scheme does not provide real-time PQ security, it enables post-mortem (PM) forgery detection via a fail-stop (FS) mechanism: computationally bounded signers (i.e., BSs) can identify and prove forgeries once the underlying assumptions are broken [50, 51]. This FS mechanism halts the system upon security breaks, minimizing damage and further exploitation. Since forgery detection relies on the integrity of audit logs, we incorporate a distributed audit logging system secured with PQ threshold signatures to support post-mortem detection (see Section 5.3.4).

Scope. Our objective is to design an authentication framework for 5G UE-BS communication. Since SIB1 conveys critical RAN information and the broadcast schedule for subsequent SIB messages, its authentication ensures that devices receive legitimate access details and scheduling. We therefore identify SIB1 as the most essential message to protect, and implement BORG primarily for its authentication. However, we identify that our mechanism is directly deployable for other SIB messages. Our scope excludes authentication and key agreement procedures (e.g., 5G-AKA [52, 53]), as BORG targets the broadcast bootstrapping plane (SIB1 authenticity and BS legitimacy) that precedes NAS/AKA, while 5G-AKA provides UE–core mutual authentication and session key establishment. Hence, BORG is orthogonal and complementary to AKA: it does not replace key agreement but ensures that the UE only initiates AKA with a verified BS. In practice, the Core Key Generator (Used in BORG) role can be realized as a logical function co-located with existing key-management entities, and the required master public key can be provisioned through standard USIM/eSIM updates. Also, BORG can be integrated with the AKA procedure through a policy gate, ensuring that the AKA process is initiated only after a fresh BORG-verified SIB1 has been received. This linkage prevents fake BS–driven bootstrapping while preserving the cryptographic structure of AKA. Similarly, side-channel attacks, including physical key extraction, are also out of scope. Additionally, our framework does not address UE-to-BS privacy, denial of service, passive eavesdropping, jamming, overshadowing, or other physical-layer attacks, which require separate, orthogonal defenses on other layers of 5G such as physical-layer encryption or anti-jamming mechanisms.

3.2. Security Model

Following the hierarchical IBS [54, 39] and Schnorr-based (threshold) signature security models [27, 8], we define the Existential Unforgeability under a selective-ID, adaptive Chosen Message-and-ID Attack (EUF-sID-CMIA) for a HITFS scheme through a game between the adversary $\mathcal{A}$ and challenger $\mathcal{C}$. The adversary $\mathcal{A}$ controls fewer than t signing participants and has access to the following oracles: *(i)* Key Extraction Oracle $\mathcal{O}_E$: Given a user ID at level k with n users, it returns the secret key shares $\{sk_{ID_{k,i}}\}_{i=1}^{n}$. *(ii)* Preprocessing Oracle $\mathcal{O}_P$: On input signing round j and user identity ID, it provides the commitment values for signing. *(iii)* Signing Oracle $\mathcal{O}_S$: Given a message m and user ID, it executes the signing procedure and returns a valid signature σ. *(iv)* Random Oracles $\mathcal{O}_{H_1}$ and $\mathcal{O}_{H_2}$: Queries to hash functions H_1 and H_2 are modeled as interactions with a random oracle, an idealized black-box that returns truly random outputs for each unique query while maintaining consistency across repeated inputs.

Definition 3.1. *The EUF-sID-CMIA security experiment $Expt^{\text{EUF-sID-CMIA}}_{\texttt{HITFS}}$ for a HITFS signature scheme is defined as follows:*

- *$\mathcal{C}$ runs $\texttt{HITFS.Setup}(1^\kappa)$ and returns the PK_{ID_0} and the public parameters to the adversary $\mathcal{A}$.*
- $(m^*, \vec{ID}_k^*, \vec{Q}_{ID_k^*}, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_E, \mathcal{O}_P, \mathcal{O}_S}(PK_{ID_0}, params)$

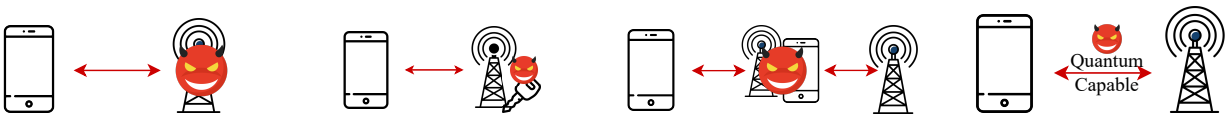
$\mathcal{A}$ wins the experiment if the forged signature passes verification ($1 \leftarrow \texttt{MVerify}(m^*, \vec{ID}_k^*, \vec{Q}_{ID_k^*}, \sigma^*)$) while satisfying the following conditions: *(i)* The target ID^* or any of its prefixes was not queried to $\mathcal{O}_E$. *(ii)* The commitment values used for signing were not queried to the preprocessing oracle $\mathcal{O}_P$. *(iii)* The message-and-ID pair $(m^*, \vec{ID}')$, where $\vec{ID}'$ is a prefix of ID^*, was not queried to $\mathcal{O}_S$. *(iv)* The PoF(.) or hash queries H_1 on the secret random value α_i (for $i \in \{0, k\}$) were not invoked during the security experiment. The forger's advantage in winning the game is defined as $Pr[Expt^{\text{EUF-sID-CMIA}}_{\text{HITFS}}(\mathcal{A}) = 1]$.

Following the principles of fail-stop signature schemes [50, 51], we formalize the security of a HITFS scheme through the following properties: *(i) Signer-Side Security*, which ensures that a quantum-capable adversary controlling fewer than t signers cannot produce an undetectable forgery; *(ii) Verifier-Side Security*, which guarantees existential unforgeability under a selective-ID, adaptive chosen message-and-ID attacks (*EUF-sID-CMIA*, Definition 3.1); and *(iii) Non-Repudiation*, which prevents signers from falsely denying valid signatures. These properties are quantified by distinct security parameters: λ_1 for signer-side fail-stop security, λ_2 for non-repudiation, and κ for verifier-side unforgeability [41, 55, 56].

Definition 3.2. *A HITFS provides λ_1-bit signer-side fail-stop security if, for any quantum-capable adversary $\mathcal{A}$ controlling fewer than t-out-of-n signers, the following holds:*

$$\Pr\begin{bmatrix} 1 \leftarrow \texttt{HITFS.MVerify}(m^*, \vec{ID}_k, \vec{Q}_{ID_k}, \sigma_k^*) \wedge \\ 0 \leftarrow \texttt{HITFS.PoF}(\{\hat{e}_{i,j}^*\}_{i=1}^{t}, \{\hat{d}_{i,j}^*\}_{i=1}^{t}, m^*, \sigma_k^*, hist) \end{bmatrix} \leq negl(\lambda_1)$$

where the forged signature passes the verification, and HITFS.PoF is the proof generated by at least one honest signer. This bound holds as long as $\mathcal{A}$ has not queried the



(a) Fake Base Stations (b) Compromised Key (c) MiTM Attacker (d) Quantum Adversary

Figure 3: Outline of Our Threat Models.

signing oracle on m^* nor obtained the commitment values $(\{\hat{e}^*_{i,j}\}^t_{i=1}, \{\hat{d}^*_{i,j}\}^t_{i=1})$ from uncorrupted signers.

Definition 3.3. *A HITFS scheme provides λ_2-bit non-repudiation FS security if, for any quantum-capable $\mathcal{A}$ controlling one-out-of-t participating signers, the following holds:*

$$\Pr\left[\begin{array}{l} 1 \leftarrow \mathit{HITFS.MVerify}(m, \vec{ID}_k, \vec{Q}_{ID_k}, \sigma_k) \wedge \\ \pi^* \leftarrow \mathit{HITFS.PoF}(\{\hat{e}^*_{i,j}\}^t_{i=1}, \{\hat{d}^*_{i,j}\}^t_{i=1}, m, \sigma^*_k, hist) \end{array}\right] \leq negl(\lambda_2)$$

where σ_k is generated according to the signing protocol, and π^* is the forgery proof which is not equal to "Not A Forgery". This bound holds as long as $\mathcal{A}$ has not queried the commitment values $(\{\hat{e}^*_{i,j}\}^t_{i=1}, \{\hat{d}^*_{i,j}\}^t_{i=1})$ of any uncorrupted signers.

4. Feasibility Analysis of NIST-PQC for Initial Bootstrapping in 5G Cellular Network

4.1. Challenges in Deploying NIST-PQC Signatures

For any BS authentication mechanism, it is critical that the BS signs the SIB1 message and, ideally, embeds the signature within the same packet to avoid additional overhead [2]. However, including the signature in SIB1 is challenging due to strict size constraints. According to the 3GPP RRC specification [21], the maximum allowable size for the SIB1 message is 2976 bits or 372 bytes (section 5.2.1). However, for initial communication, not all the bytes of the SIB1 are used. For instance, a minimally configured SIB1 typically occupies 80–100 bytes. To assess real-world usage and configuration, we take measurements for SIB1 messages from two major U.S. vendors across multiple BSs–covering 8 different physical cells. The observed SIB1s had a minimum size of 108 bytes and a maximum of 120 bytes. These findings suggest that, under current commercial vendor deployments, compact signatures can be piggybacked without requiring newly introduced messages. In contrast, larger signatures would require further modifications to network operations (see Section 4.2).

Thus, to avoid fragmentation, the signature must fit within the current packet size alongside its standard fields. Unfortunately, NIST-selected signatures produce signature sizes that exceed this limit and are incompatible with the 372-byte maximum size of SIB1. For example, the *FN-DSA* [57] yields a 1280-byte signature and a 1793-byte public key at NIST security level 5, and even at level 1, it requires 666 bytes for the signature and 897 bytes for the public key. Similarly, *SLH-DSA* [58] produces a signature size of nearly 17 KB at level 3, making it highly impractical for 5G UE-to-BS communication. The *ML-DSA* signature (e.g., *Dilithium2* [36]) produces a 2420-byte signature and a 1312-byte public key. These sizes far exceed the SIB1 limit, and the only way to transmit such signatures would be to fragment both the signature and public key across multiple SIB1 packets.

4.2. Fragmentation Constraints

Specifically, to transmit the additional 3732 bytes required by *Dilithium2* (2420-byte signature and 1312-byte public key), and assuming 290 bytes of free space per SIB1, the BS would need to send 13 separate SIB1 packets, each containing a fragment of the signature-key pair. The UE must then extract these fragments and reconstruct the full signature and key for verification. Notably, this overhead accounts for only a single level in the certificate chain; transmitting longer chains would exacerbate the problem. The resulting communication and processing burden on both ends substantially increases the setup-to-authentication time, making this approach impractical for 5G environments.

We have already established the communication overhead of broadcasting large keys or certificates by fragmenting them across multiple SIB1 packets. However, the latency implications further exacerbate this challenge. According to the RRC specification, "*The SIB1 is transmitted on the DL-SCH with a periodicity of* 160 *ms and variable transmission repetition periodicity within 160 m. The default transmission repetition periodicity of SIB1 is* 20 *ms but the actual transmission repetition periodicity is up to network implementation.*" (section 5.2.1 [21]). This implies a default delay of 20 ms between consecutive SIB1 packet transmissions and maximum delay of 160 ms, even when broadcasting identical network parameters with different signature fragments. Consequently, transmitting these fragments introduces a baseline latency ranging from $20 \times 12 = 240$ ms upto $160 \times 12 = 1920$ ms, not including the time for packet generation at the BS and reconstruction at the UE. Moreover, the delivery of SIB1 can be unreliable over the DL-SCH channel. Therefore, each packet must be assigned a sequence number for identification. Now, in this unreliable scenario, even with the newly introduced sequence number, we need to consider the situation when the UE receives out-of-order packets and must wait for all 13 packets on its end before reconstructing the key or signature. In the worst case, the UE can receive the 2nd packet first, followed by the 3rd, and so on; until it receives the 1st packet, followed by the 2nd; up to the 13th $(p_2, p_3, \ldots, p_{13}, p_1, p_2, \ldots p_{12}, p_{13})$. Thus, under a uniformly random packet delivery model, the expected number of packets needed for successful key/signature delivery rises to $13 + \lfloor 13/2 \rfloor = 19$, resulting in

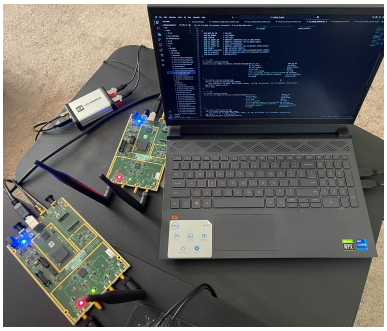


Figure 4: Testbed setup for 5G end-to-end communication. The gNB (USRP: bottom-left) and UE (USRP: top-right) are connected to a laptop and GPSDO.

an overall transmission delay ranging from $20\times18 = 360$ ms upto $160 \times 18 = 2880$ ms in the DL-SCH, rendering this approach unsuitable for time-sensitive 5G authentication.

To handle the fragmentation scheme with a potential out-of-order packet arrival scenario, one needs to consider a sliding window process to keep track of the valid in-sequence SIB1s. The sliding window process can sort the incoming packets according to their sequence numbers. When all the required packets (in our ML-DSA single-chain example, there will be 13 packets in the best case and 25 packets in the worst case) are received, the process merges the extracted fragments to reconstruct the key/signature. In addition to the above challenges, transmitting multiple SIB1 packets adds significant complexity to lower protocol layers. For example, in the srsRAN open-source 5G stack, even when multiple SIB1 messages are generated, the MAC layer permits only a single packet for scheduling at a time. Moreover, the protocol expects each BS to broadcast one SIB1 message, after which the UE initiates connection procedures. These limitations further underscore the impracticality of existing PQC solutions in the context of 5G BS authentication.

4.3. Comparative Analysis on 5G Testbed

Over-the-air Testbed Setup. To understand the applicability of various cryptographic solutions for 5G UE authentication, we set up a Software Defined Radio (SDR)-based testbed and run the algorithms for over-the-air communication. We use srsRAN and open5GS open-source implementations for this purpose. More specifically, we run srsUE on a USRP *B*210 and srsgNB on another USRP *B*210–both connected to the same computer through USB 3.0 ports. We also use a Leo Bodnar GPSDO to ensure a seamless 10 MHz external clock for the USRP devices. The attached srsgNB to the open5GS core gets connected to the srsUE through over-the-air communication. Since the baseband of commercial UEs cannot be modified, we adopt a best-effort approach and conduct our analysis using a widely used srsRAN-Open5GS testbed. Our setup is consistent with existing works on 4G/5G bootstrapping [4, 2]. Fig. 4 shows our complete testbed setup.

We evaluate existing schemes alongside our solution, BORG, in an over-the-air setup, measuring both cryptographic and network-induced delays. For *ML-DSA* with two certificate chains where fragmentation becomes essential, authentication requires 33 additional packets, introducing roughly 12 *KB* of overhead. In contrast, BORG requires no additional packets, making it highly compatible with current 5G protocol constraints. Also, *ML-DSA* incurs a latency of approximately ranging from 662.47 ms upto 5282.47 ms, which is nearly $221 \sim 1767\times$ higher than that of BORG (see TABLE 2), rendering it impractical. While schemes like *EC-Schnorr* avoid fragmentation, they lack PQ forgery detection and compromise resilience. This puts BORG in a suitable spot of compatibility both from a PQ and 5G network perspective. Further performance details are provided in Section 6.

5. The Proposed Future-Proof Solution

We begin with an overview of BORG, followed by algorithm descriptions and protocol instantiation for 5G networks.

5.1. The Proposed BORG Scheme

Given the infeasibility of current NIST-PQC standards, BORG focuses on conventional secure techniques enhanced with key features: threshold signing for distributed trust, IBS to lift certificate burdens, and fail-stop mechanisms with PQ threshold audit logging for post-mortem PQ forgery detection. We present BORG in Algorithms 1-4.

Algorithm 1 BORG *(Setup and Key Extraction)*

$(sk_{ID_0}, PK_{ID_0}, params) \leftarrow$ BORG.Setup(1^κ): CKG runs this algorithm once to set up the system.

1: $\alpha_0 \xleftarrow{\$} \mathbb{Z}_q$, $sk_{ID_0} \leftarrow H_1(\alpha_0)$, and $PK_{ID_0} \leftarrow g^{sk_{ID_0}} \mod p$
2: **return** sk_{ID_0}, PK_{ID_0}, and $params \leftarrow \{p, q, H_1, H_2\}$

$(\{sk_{ID_{k,i}}\}_{i=1}^n, \{PK_{ID_{k,i}}\}_{i=1}^n, \vec{Q}_{ID_k}) \leftarrow$ BORG.Extract$(ID_k, \vec{Q}_{ID_{k-1}}, sk_{ID_{k-1}})$: This algorithm is run by the user at level $(k-1)$ to generate key pairs for users at level k.

1: $\alpha_k \xleftarrow{\$} \mathbb{Z}_q$, $r_k \leftarrow H_1(\alpha_k)$, and $Q_{ID_k} \leftarrow g^{r_k} \mod p$
2: $\vec{Q}_{ID_k} \leftarrow (\vec{Q}_{ID_{k-1}}, Q_{ID_k})$ and $h_{ID_k} \leftarrow H_1(ID_k || \vec{Q}_{ID_k})$
3: $sk_{ID_k} \leftarrow sk_{ID_{k-1}} \cdot h_{ID_k} + r_k \mod q$
4: $f(x) = sk_{ID_k} + \sum_{i=1}^{t-1} a_i \cdot x^i \mod q$, where $a_i \xleftarrow{\$} \mathbb{Z}_q$, $i \in \{1, ..., t-1\}$
5: **for** $i = 1, 2, \ldots, n$ **do**
6: $sk_{ID_{k,i}} \leftarrow f(i)$ and $PK_{ID_{k,i}} \leftarrow g^{sk_{ID_{k,i}}} \mod p$
7: **return** $(\{sk_{ID_{k,i}}\}_{i=1}^n, \{PK_{ID_{k,i}}\}_{i=1}^n, \vec{Q}_{ID_k})$

BORG.Setup (Algorithm 1) initializes the system by generating the master secret and public keys (sk_{ID_0}, PK_{ID_0}), and publishing the public parameters *params*. BORG.Extract (Algorithm 1) enables hierarch-ical key extraction, allowing each level to derive key pairs for the subsequent level. Specifically, users at level $k-1$ utilize their secret key $sk_{ID_{k-1}}$, the public key vector $\vec{Q}_{ID_{k-1}} = \{PK_{ID_0}, Q_{ID_1}, \ldots,$

$Q_{ID_{k-1}}\}$, and the group identity ID_k of the lower level to compute the group verification key Q_{ID_k} and derive individual key pairs $(sk_{ID_{k,i}}, PK_{ID_{k,i}})$ for each user $i = 1, \dots, n$ at level k. Secret keys are then securely distributed to each user. The derived secret key follows a Schnorr signature structure and can be reconstructed via Lagrange interpolation [59] from any t-out-of-n users at level k: $sk_{ID_k} = \sum_{i=1}^{t} \lambda_i \cdot sk_{ID_{k,i}} \ mod \ q$.

Algorithm 2 BORG *(Preprocessing)*

$\mathcal{L}_i \leftarrow$ BORG.Preprocess(J): n users at level k execute this algorithm to generate commitment values, enabling them to sign up to J messages.

for $i = 1, 2, \dots, n$ **do**
  for $j = 1, 2, \dots, J$ **do**
    $(\hat{e}_{i,j}, \hat{d}_{i,j}) \xleftarrow{\$} \mathbb{Z}_q \times \mathbb{Z}_q$
    $e_{i,j} \leftarrow H_1(\hat{e}_{i,j}||j||ID_{k,i})$, $d_{i,j} \leftarrow H_1(\hat{d}_{i,j}||j||ID_{k,i})$
    $E_{i,j} \leftarrow g^{e_{i,j}} \mod p$ and $D_{i,j} \leftarrow g^{d_{i,j}} \mod p$
  Send $(\{E_{i,j}\}_{j=1}^{J}, \{D_{i,j}\}_{j=1}^{J})$ to the other $n-1$ users.
for $i = 1, 2, \dots, n$ **do**
  if $(\{E_{i,j}\}_{j=1}^{J}, \{D_{i,j}\}_{j=1}^{J}) \in \mathbb{G}$ **then**
    $\mathcal{L}_{i,j} \leftarrow (E_{i,j}, D_{i,j})$
  return $\mathcal{L}_i \leftarrow (i, \{E_{i,j}\}_{j=1}^{J}, \{D_{i,j}\}_{j=1}^{J})$

Akin to Schnorr-style threshold signatures (i.e., [27]), the signing follows a two-round protocol with three phases: (i) preprocessing: to generate a shared list of random commitments values, (ii) threshold signing: allows participants to compute their signature shares, and (iii) aggregation: to combine these shares into a group signature. The BORG.Preprocess phase (Algorithm 2) is jointly executed by all n users at level k to generate random commitment values required for signing up to J messages. Each user $i \in \{1, \dots, n\}$ computes commitment pairs $(E_{i,j}, D_{i,j})$ for $j = 1$ to J, and shares them with the others. After verifying their validity, users append the values to the commitment list $\mathcal{L}_{i,j}$, resulting in a consistent finalized list $\mathcal{L}_i$ across all participants. In practice, $\mathcal{L}_i$ may be published at a predefined location (e.g., a public bulletin) accessible to all users.

In BORG.Sign (Algorithm 3), each signer uses its secret key share $sk_{ID_{k,i}}$ and the commitment list $\mathcal{L}_i$ to compute a signature share for message m at index $j \in \{1, \dots, J\}$, and submits it to the other β signing participants. The signer set size β is predetermined ($t \leq \beta \leq n$) and must meet the threshold t to produce a valid group signature. Upon receiving $\beta - 1$ shares, each signer verifies them individually (aborting on failure) and aggregates the valid shares into the group signature $\sigma_{k,j} = (R_j, z_j)$. The BORG.MVerify (Algorithm 3) follows the standard Schnorr signature verification process. Using the vector of public keys $\vec{Q}_{ID_k}$ and identities $\vec{ID}_k$, the verifier validates the signature $\sigma_{k,j}$ on the message m_j. The correctness of the verification algorithm resembles Schnorr-based signature verification [8] and is given in Section A.2.

Multiple valid signatures may satisfy the verification algorithm; however, even if a capable adversary obtains such signatures, they cannot distinguish those genuinely generated by authorized signers [41]. Leveraging this principle and following fail-stop mechanisms (e.g., [41, 42]), signers at level k will claim forgery by invoking the forgery proof algorithm BORG.PoF, while higher-level authorities at level $k-1$ run BORG.PoFVerify to validate the claim.

Algorithm 3 BORG *(Message Signing and Verification)*

$\sigma_{k,j} \leftarrow$ BORG.Sign($m_j, \mathcal{L}_i, \{sk_{k,i}\}_{i=1}^{\beta}$): At level k, β participants ($\beta \in [t, n]$) execute this algorithm:

for $i = 1, 2, \dots, \beta$ **do**
  $\rho_{i,j} \leftarrow H_1(i||m_j||\{\mathcal{L}_{i,j}\}_{i=1}^{\beta})$
  $R_{i,j} \leftarrow D_{i,j} \cdot (E_{i,j})^{\rho_{i,j}} \mod p$
  $R_j \leftarrow \prod_{i=1}^{\beta} R_{i,j} \mod p$ and $h_j \leftarrow H_2(R_j||Q_{ID_k}||m_j)$
  $z_{i,j} \leftarrow d_{i,j} + e_{i,j} \cdot \rho_{i,j} + \lambda_i \cdot sk_{ID_{k,i}} \cdot h_j \mod q$
Send $\{z_{i,j}\}_{i=1}^{\beta}$ to $\beta - 1$ participants.
Each participant i performs:
for $i = 1, 2, \dots, \beta$ **do**
  $\rho_{i,j} \leftarrow H_1(i||m_j||\{\mathcal{L}_{i,j}\}_{i=1}^{\beta})$
  $h_j \leftarrow H_2(R_j||Q_{ID_k}||m_j)$
  if $g^{z_{i,j}} \neq R_{i,j} \cdot PK_i^{\lambda_i \cdot h_j} \mod p$ **then**
    return $\perp$.
  else $z_j \leftarrow \sum_{i=1}^{\beta} z_{i,j} \mod q$ and $R_j \leftarrow \prod_{i=1}^{\beta} R_{i,j} \mod p$
    return $\sigma_{k,j} \leftarrow (R_j, z_j)$

$\{0, 1\} \leftarrow$ BORG.MVerify($m_j, \vec{ID}_k, \vec{Q}_{ID_k}, \sigma_{k,j}$): This algorithm is executable by any user within the network.

$h_{ID_\ell} \leftarrow H_1(ID_\ell||\vec{Q}_{ID_\ell})$ for $\ell = 1, 2, \dots, k$
$Q \leftarrow \prod_{\ell=1}^{k-1} (Q_{ID_\ell})^{\prod_{\omega=\ell+1}^{k} h_{ID_\omega}}$
$h_j \leftarrow H_2(R_j||Q_{ID_k}||m_j)$
if $g^{z_j} \stackrel{?}{=} R_j \cdot (Q \cdot Q_{ID_k} \cdot (PK_{ID_0})^{\prod_{\ell=1}^{k} h_{ID_\ell}})^{h_j} \mod p$ **then**, **return** 1

BORG.PoF (Algorithm 4) is invoked by signers at level k upon detecting a suspected forgery of message m_j (e.g., via signature audit logs). Using the signature history (*hist*) to identify the message index j, the β participating signers reveal their secret nonces $(\{\hat{e}_{i,j}\}_{i=1}^{\beta}, \{\hat{d}_{i,j}\}_{i=1}^{\beta})$ originally generated during preprocessing. Each signer reconstructs the signature component R_j as in the signing process and compares it with the corresponding component in the suspected signature σ'_k. If they match, the signer outputs π as "*Not a Forgery*"; otherwise, it outputs the proof $\pi = (j, \{\hat{e}_{i,j}\}_{i=1}^{\beta}, \{\hat{d}_{i,j}\}_{i=1}^{\beta})$, which is submitted to the higher-level authority at level $k-1$. BORG.PoFVerify (Algorithm 4) is executed by the level $k-1$ authority. Using the secret value α_k chosen during key extraction, the verifier first checks the validity of the public and group verification keys. If this fails, the proof is rejected. Otherwise, it reconstructs R_j from the disclosed nonces in π and compares it with R'_j from the suspected forged signature. A match results in rejection of the forgery claim; a mismatch confirms forgery, prompting system halt due to a detected security breach.

Algorithm 4 BORG *(Forgery Detection and Verification)*

$\pi \leftarrow$ BORG.PoF$(\{\hat{e}_{i,\mathrm{j}}\}_{i=1}^{\beta}, \{\hat{d}_{i,\mathrm{j}}\}_{i=1}^{\beta}, m_{\mathrm{j}}, \sigma'_k, hist)$: This algorithm is run by the β signing participants at level k to prove the forgery of a signature σ'_k to entities at level $k-1$:

j $\leftarrow hist$ and $(R'_{\mathrm{j}}, z'_{\mathrm{j}}) \leftarrow \sigma'_k$
for $i = 1, 2, \ldots, \beta$ **do**
  Reveal $(\hat{e}_{i,\mathrm{j}}, \hat{d}_{i,\mathrm{j}})$ to other $\beta - 1$ participants

Each participant i performs:

for $i = 1, 2, \ldots, \beta$ **do**
  $e_{i,\mathrm{j}} \leftarrow H_1(\hat{e}_{i,\mathrm{j}}||\mathrm{j}||m_{\mathrm{j}})$ and $d_{i,\mathrm{j}} \leftarrow H_1(\hat{d}_{i,\mathrm{j}}||\mathrm{j}||m_{\mathrm{j}})$
  $E_{i,\mathrm{j}} \leftarrow g^{e_{i,\mathrm{j}}} \mod p$ and $D_{i,\mathrm{j}} \leftarrow g^{d_{i,\mathrm{j}}} \mod p$
  $\rho_{i,\mathrm{j}} \leftarrow H_1(i||m_{\mathrm{j}}||\{\mathcal{L}_{i,\mathrm{j}}\}_{i=1}^{\beta})$
$R_{\mathrm{j}} \leftarrow \prod_{i=1}^{\beta} D_{i,\mathrm{j}} \cdot (E_{i,\mathrm{j}})^{\rho_{i,\mathrm{j}}} \mod p$
if $R'_{\mathrm{j}} = R_{\mathrm{j}}$ **then**
  return $\pi \leftarrow$ "*Not A Forgery*"
if $R'_{\mathrm{j}} \neq R_{\mathrm{j}}$ **then**
  return $\pi \leftarrow (\mathrm{j}, \{\hat{e}_{i,\mathrm{j}}\}_{i=1}^{\beta}, \{\hat{d}_{i,\mathrm{j}}\}_{i=1}^{\beta})$

$\{0,1\} \leftarrow$ BORG.PoFVerify$(\alpha_k, sk_{ID_{k-1}}, \vec{Q}_{ID_k}, m, \sigma'_k, \pi)$: This algorithm is run by the level $k-1$ to verify the proof of forgery.

if $\pi' = \}\}$*Not A Forgery*ε **then**
  return $\perp$
if $\pi' = (\mathrm{j}, \{\hat{e}_{i,\mathrm{j}}\}_{i=1}^{\beta}, \{\hat{d}_{i,\mathrm{j}}\}_{i=1}^{\beta})$ **then**
  $r_k \leftarrow H_1(\alpha_k)$ and $Q_{ID_k} \leftarrow g^{r_k} \mod p$
  $h_{ID_k} \leftarrow H_1(ID_k||\vec{Q}_{ID_k})$
  for $i = 1, 2, \ldots, \beta$ **do**
    if $\prod_{i=1}^{\beta} PK_{ID_{k,i}}^{\lambda_i} \neq (g^{h_{ID_k} \cdot sk_{ID_{k-1}}}) \cdot Q_{ID_k}$ **then**
      return $\perp$
    else $(R'_{\mathrm{j}}, z'_{\mathrm{j}}) \leftarrow \sigma'_k$
  for $i = 1, 2, \ldots, \beta$ **do**
    $e_{i,\mathrm{j}} \leftarrow H_1(\hat{e}_{i,\mathrm{j}}||\mathrm{j}||m_{\mathrm{j}})$ and $d_{i,\mathrm{j}} \leftarrow H_1(\hat{d}_{i,\mathrm{j}}||\mathrm{j}||m_{\mathrm{j}})$
    $E_{i,\mathrm{j}} \leftarrow g^{e_{i,\mathrm{j}}} \mod p$ and $D_{i,\mathrm{j}} \leftarrow g^{d_{i,\mathrm{j}}} \mod p$
    $\rho_{i,\mathrm{j}} \leftarrow H_1(i||m_{\mathrm{j}}||\{\mathcal{L}_{i,\mathrm{j}}\}_{i=1}^{\beta})$
  $R_{\mathrm{j}} \leftarrow \prod_{i=1}^{\beta} D_{i,\mathrm{j}} \cdot (E_{i,\mathrm{j}})^{\rho_{i,\mathrm{j}}} \mod p$
  if $R'_{\mathrm{j}} = R_{\mathrm{j}}$ **then**, **return** 0
  else, **return** 1

5.2. Security Analysis

Following Definition 3.1, we prove $EUF\text{-}sID\text{-}CMIA$ security of BORG under the hardness of the (Elliptic Curve)-DLP based on the generalized forking lemma in the random oracle model [60, 61]. Additionally, based on Definition 3.2, we prove the signer-side fail-stop security of BORG under the hardness of second preimage resistance of a cryptographically secure hash function.

Theorem 1. *If an adversary $\mathcal{A}$ can $(q_E, q_P, q_S, q_{H_1,H_2})$-break BORG in the random oracle model (Definition 3.1) with an advantage ϵ in time τ while having access to at most $(t-1)$-out-of-n signing participants, where q_E, q_P, q_S, q_{H_1}, and q_{H_2} denote queries to key extraction, preprocessing, signing, and hash functions H_1 and H_2, then an algorithm $\mathcal{C}$ can be constructed to break the (EC)DLP in group $\mathbb{G}$.*

Theorem 2. *BORG provides λ_1-bit signer-side fail-stop security against quantum-capable adversaries controlling up to $(t-1)$-out-of-n signing participants, as formalized in Definition 3.2, and λ_2-bit non-repudiation security against quantum adversaries with access to one-out-oft signing participants, as captured in Definition 3.3. Both guarantees rely on the hardness of breaking the second preimage resistance of a cryptographically secure hash function, while providing κ bit verifier-side security via EUF-sID-CMIA in the random oracle model as defined in Definition 3.1.*

Full security proofs are presented in Appendix A.3.

5.3. Instantiation of BORG for 5G Network

This section outlines the instantiation of the BORG algorithm for 5G and beyond mobile networks, providing a high-level overview while detailing each step in Figure 5.

Solution Architecture. The proposed architecture adopts a two-layer design comprising: *(1)* a newly introduced Core Key Generator (CKG) for key generation and system initialization within the 5G core; *(2)* the Access and Mobility Management Function (AMF); *(3)* a set of Base Stations (BSs); and *(4)* User Equipment (UE), representing end-user devices such as smartphones, laptops, and IoT nodes.

Our Authentication Protocol Overview: The CKG manages key setup for the AMF, which in turn handles key extraction for all BSs. Key extraction occurs periodically. Each entity is identified by an ID concatenated with a secret share expiry timestamp, used as input in key extraction. CKG key pairs, valid for years, are pre-installed in the UE's USIM. BS keys default to a one-day validity, adjustable as needed. Due to physical exposure, BSs benefit from the (t, n) threshold, requiring compromise of multiple entities, making frequent updates (e.g., hourly) both effective and practical. These periods are configurable by the 5G core.

Key extraction is efficient for both AMF and CKG, scaling efficiently even at a global level. The derived secret keys are securely distributed from the AMF to the BSs using authenticated control channels. In practice, this can be achieved via the Xn Application Protocol (XnAP)—which supports secure inter-gNB signaling—or through out-of-band non-3GPP access mechanisms similar to those used for eSIM remote provisioning, ensuring mutual authentication, confidentiality, and integrity during key transfer [62]. Similarly, Base Station IDs can be distributed using the Protocol Extension Container Information Element (*ProtocolExtensionContainer IE*) available in the XnAP messages. This IE is particularly kept to carry additional elements for communication, all while keeping backward compatibility. The (t, n) threshold ensures that at least t BSs collaborate to generate a valid signature, with nonces preprocessed in batches to further reduce costs. All participating BSs also sign the resulting SIB signature using their ThPQ secret shares and record the final audit signature in distributed cold storage [63, 64, 65]. Since each BS holds the aggregated SIB signature and independently logs it using the (t', n) threshold audit signature (where $t \leq t'$) via the ThPQ scheme, we eliminate any single point of failure. This approach offers a scalable solution for long-term archival in large-scale 5G deployments and enables fault-tolerant PQ threshold audit logging to support PM forgery detection.

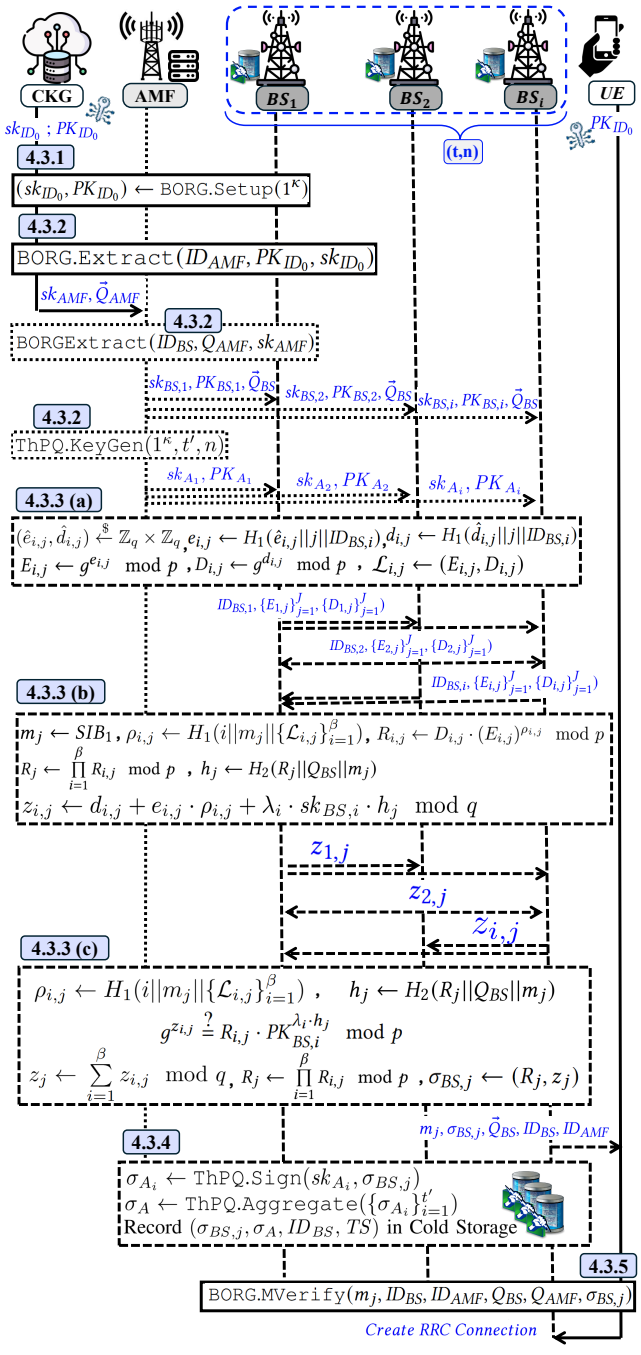


Figure 5: Our protocol for authenticating 5G cellular BSs

The UE verifies the SIB1 signature using only the pre-installed master key, with other public key data sent alongside the signature, ideal for resource-constrained devices. In the event of forgery, BSs disclose their nonces to generate a proof, which is sent to the AMF, enabling a system halt if needed. Precisely, the FS mechanism allows the core network to isolate compromised BSs and suspend authentication, preventing forged SIBs from spreading. Upon provable detection, the system halts affected operations, enabling swift, policy-enforced recovery with minimal disruption.

5.3.1. *System Initialization Phase*

In the two-layered architecture, the CKG handles the one-time system setup during initial mobile network deployment by executing `BORG.Setup` as defined in Algorithm 1.

5.3.2. *Key Extraction Phase*

This phase initiates with the CKG running `BORG.Extract` (Algorithm 1) to derive the AMF's key pair $(sk_{AMF}, \vec{Q}_{AMF})$ from the master secret sk_{ID_0}, which is then securely transmitted to the AMF. The AMF then applies the same procedure to generate threshold-shared keys for the BSs. Under a (t, n) configuration (e.g., 2-of-3), each BS_i receives a share $sk_{BS,i}$ and public key $PK_{BS,i}$, with all BSs sharing a group verification key $\vec{Q}_{BS}$. Any t out of n can jointly produce a valid signature verifiable by $\vec{Q}_{BS}$. Additionally, the AMF executes `ThPQ.KeyGen` to produce the audit public key PK_A and secret key shares $(sk_{A_1}, \ldots, sk_{A_n})$, enabling any $t' \geq t$ BSs to generate a valid `ThPQ` signature for secure audit logging. All keys are distributed over secure channels.

5.3.3. *Signing Broadcast Messages Phase*

To sign SIB1, $\beta \in [t, n]$ BSs collaboratively generate a group signature through three phases: *(a) Preprocessing:* As defined in `BORG.Preprocess` (Algorithm 2), this phase is precomputed for a given window (e.g., daily). Each BS_i uses its NRCell_ID ($ID_{BS,i}$) to generate random nonces and compute commitment pairs $(\{E_{i,j}\}_{j=1}^{J}, \{Di, j\}_{j=1}^{J})$ for J messages. These are exchanged, verified, and appended to local commitment lists $\mathcal{L}_{i,j}$, yielding a consistent list $\mathcal{L}_i$ shared across all BSs, optionally published in a predefined location (e.g., a bulletin). *(b) Threshold Signing:* Following `BORG.Sign` (Algorithm 3), each BS_i uses its key share and commitment list $\mathcal{L}_i$ to sign message $m_j \leftarrow SIB_1$, then sends the resulting signature share to the remaining $\beta - 1$ signers. The number of signers β is predetermined prior to signing. *(c) Aggregation:* Upon receiving the signature shares, each BS verifies the signature shares and aggregates the valid ones into the final group signature $\sigma_{BS,j} = (R_j, z_j)$. The BS with the strongest signal strength broadcasts the $(m_j, \sigma_{BS,j}, \vec{Q}_{BS}, ID_{BS}, ID_{AMF})$.

5.3.4. *Audit Logging Phase*

To strengthen forgery detection and independent of the 5G signing process, a subset of t' participating BSs (where $t \leq t'$) collaboratively sign the SIB1 signature $\sigma_{BS,j}$. Using `ThPQ.Sign`, each BS_i signs $\sigma_{BS,j}$ and sends the share to the other participants. Upon collecting t' valid `ThPQ` shares, each BS runs `ThPQ.Aggregate` to produce the final threshold signature σ_A. The audit log entry $(\sigma_{BS,j}, \sigma_A, ID_{BS}, TS)$ is then stored in distributed cold storage, ensuring fault-tolerant threshold auditability and and supporting proof-of-forgery verification against quantum adversaries. In practice, the distributed cold storage is realized as a lightweight, append-only audit repository replicated across trusted RAN/Core entities (e.g., AMF, gNBs), where updates occur periodically via authenticated control channels (e.g., XnAP, TLS). This design imposes

negligible signaling overhead while maintaining tamper-evident, verifiable records for post-mortem analysis.

5.3.5. *Signature Verification Phase*

The signature verification process follows BORG.MVerify (Algorithm 3). Given a signature on the SIB1 message, the group verification key Q_{BS}, the AMF's public key Q_{AMF}, and the CKG's master public key PK_{ID_0} (pre-installed on the user's device), the UE verifies the broadcast message to authenticate the BSs before initiating an RRC connection.

5.3.6. *Forgery Detection and Verification Phase*

Each BS has access to the authenticated distributed cold storage that records all SIB messages and aggregated signatures. Upon suspecting forgery, a BS initiates the PoF protocol with the other t participating signers, as illustrated in Figure 6. For a suspected forged signature σ'_{BS} on a SIB1 message, the β participating signers identify the message index j from the cold storage log and reveal their corresponding preprocessing nonces $(\hat{e}_{i,\mathrm{j}}, \hat{d}_{i,\mathrm{j}})$ for $i = 1, \ldots, \beta$. They invoke BORG.PoF (Algorithm 4) to reconstruct R_j and evaluate the validity of σ'_{BS}. Forgery verification is performed by the AMF using BORG.PoFVerify. Upon receiving the proof π from the signing BSs, the AMF verifies the identities and public keys of the involved BSs, reconstructs R_j from π, and compares it with R'_j in σ'_{BS}. A mismatch indicates a breach in the security of the authentication system and provably confirms that the underlying security assumption (i.e., ECDLP) has been broken, prompting a scoped, policy-gated response rather than an indiscriminate shutdown: the AMF isolates and re-keys the implicated (t, n) signing group, while UEs fall back to scanning for alternative authenticated BSs per the configurable behavior of Section 5.3.7, preserving availability.

Resistance to halt-induced DoS. A halt is triggered only by a **provable** forgery, i.e., a signature that satisfies BORG.MVerify yet does not match the legitimate signers' committed nonces R_j (Algorithm 4). An adversary injecting arbitrary or malformed signatures fails verification at the UE; such messages are discarded during normal SIB1 processing (Section 5.3.7) and never reach the PoF/PoFVerify path, so they cannot induce a halt. Producing a signature that *passes* MVerify while differing from the legitimate one requires solving the (EC)DLP (Theorem 1) or quantum capability; for a computationally bounded adversary, this is infeasible, leaving no low-cost path to a detection event. The only way to force a halt is therefore to perform precisely the cryptographic break that the fail-stop property is designed to expose. Furthermore, an insider controlling fewer than t signers cannot fabricate a forgery claim against an honestly generated signature: by the λ_2-bit non-repudiation guarantee (Definition 3.3, Theorem 2), a PoFVerify-accepting proof π^* for a legitimate signature would require a second preimage of H_1, and the AMF independently re-validates each claim via BORG.PoFVerify using α_k before acting. Thus, neither external injection nor insider misbehavior provides a controllable DoS trigger.

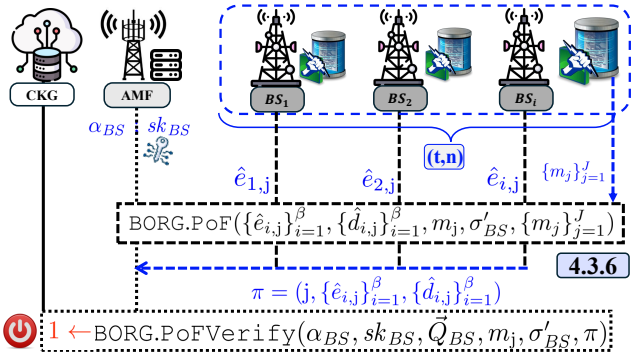


Figure 6: Instantiation of Forgery Proof and Verification

5.3.7. *Authentication failure action*

The distributed design of BORG reduces the likelihood of authentication failure. In rare cases where the UE cannot verify a BS (e.g., due to signer unavailability or missing authentication), it may continue scanning for alternative BSs. This behavior can be made configurable at the UE level, allowing users to prioritize either connectivity or security. For example, a UE may temporarily connect to an unauthenticated BS under constrained conditions while monitoring for a verifiable, authenticated BS to hand over to when available.

5.3.8. *Handling Mobility, Handover Protocol, and Roaming Scenario*

The hierarchical and distributed design of BORG enables efficient authentication in various mobility scenarios. During intra-gNB handovers, the UE can rely on previously authenticated SIBs, avoiding reauthentication. Newly issued SIBs remain verifiable using the core network's PK embedded in the authenticated structure. Also, BORG's threshold design allows uninterrupted authentication given the UE stays within range of any t out of n collaborating BSs, reducing handover latency and eliminating redundant reauthentication across adjacent BSs within the same domain. In roaming scenarios, authentication becomes more complex as the UE connects to a different network operator. BORG assumes that the UE stores the PK_{ID_0} of its home network's core-PKG within the USIM or eSIM. To authenticate SIBs from the serving network, the UE must obtain the public key of the serving network's core-PKG, distributed securely via non-3GPP access (e.g., Wi-Fi) and verified through a certificate signed by the home network's core-PKG. eSIM technology enables this secure, dynamic credential provisioning. For the roaming scenario, our approach can leverage existing telecom-level PKI frameworks such as STIR/SHAKEN [66], mandated for U.S. carriers and based on operator-issued certificates for inter-operator authentication. These nationwide deployments demonstrate the practicality of implementing our solution within today's cellular PKI ecosystem. Similar efforts are emerging globally, and 3GPP is developing complementary specifications for network PKI. While techniques, such as dynamic threshold adaptation or precomputed forward authentication

shares, may further enhance handover efficiency across networks, we leave these optimizations to future work.

5.3.9. Protection Against Relay Attacks

While BORG enables UEs to authenticate SIBs and detect fake BSs, it does not prevent relay attacks, where an adversary rebroadcasts valid SIBs from a legitimate BS at higher signal strength to mislead UEs. Since relayed messages remain valid, verification alone is insufficient. Mitigating such attacks would require distance-bounding protocols, which demand substantial changes to cellular standards and hardware. To mitigate relay attacks without overhauling existing protocols, we extend BORG with a time-bounded authentication mechanism. Each BS signs SIBs with a timestamp and short validity period, determined using transmission delay (reflecting propagation and processing time) and cryptographic delays (reflecting signing time) stored in a secure lookup table. The UE verifies message freshness by checking the timestamp against the current time, discarding any expired messages. BORG's threshold design strengthens this defense by requiring timely inputs from multiple BSs, making it difficult for an attacker to relay a complete, valid message within the allowed window.

6. Performance Evaluation

This section evaluates and compares BORG with alternative authentication schemes for 5G initial bootstrapping.

6.1. Configuration and Experimental Setup

Hardware: We assessed the efficiency of BORG protocol utilizing a standard desktop equipped with an 12^{th} Gen Intel Core $i7-12700H$@3.50 GHz, 16 GiB RAM, a 512 GiB SSD, and Ubuntu 22.04.4 LTS. The 5G testbed setup follows the configuration in Section 4. Real network packets were investigated using the Network Signal Guru Android app installed on a OnePlus Nord 5G smartphone [67].

Libraries: We employed OpenSSL library[3] for cryptographic primitives such as hash functions and EC operations (e.g., point multiplication, modular arithmetic), the Open Quantum-Safe library[4] for NIST-PQC schemes, the Ringtail library[5] for the ThPQ, and the blst[6] library for the *BLS* signature.

Parameter Selection: We configured the classical security level to 128 bits, following NIST recommendations, and the post-quantum security to NIST Level I [68]. NIST Level I provides quantum resistance approximately equivalent to 128-bit classical security. All cryptographic operations of the elliptic curve were performed over the standard curve *secp224k1*, defined on a 224-bit prime field, with *SHA-256* used as the cryptographic hash function.

srsRAN Configuration: We observe that for the first SIB1 message, the srsRAN gNB utilizes 79 bytes out of the allowed 372 bytes. Accordingly, all subsequent evaluations report the computational and communication overhead for signing a 79-byte SIB1 message, as reflected in the tables. Note that even considering slightly larger SIB1s observed from real networks, our results remain consistent.

[3] https://openssl-library.org/
[4] https://openquantumsafe.org/
[5] https://github.com/daryakaviani/ringtail
[6] https://github.com/Chia-Network/bls-signatures

6.2. Evaluation Metrics

Quantitative metrics include computational costs (e.g., signing, verification), 5G processing, cryptographic overhead (signature and key sizes), communication overhead (e.g., for transmission over-the-air), and end-to-end (E2E) delay. The 5G processing latency captures the time network entities (e.g., gNBs and UEs) spend handling cryptographic material in packets, excluding cryptographic computations, and includes processing of signature fragments when applicable. Qualitative evaluation focuses on system architecture, accountability, and breach resiliency.

6.3. Selection Criteria for Comparison Baselines

For PQ counterparts, we consider NIST-PQC signatures like lattice-based *ML-DSA* [36], *FN-DSA* [57], and hash-based *SLH-DSA* [58]. Given the very large signature and execution times of hash-based alternatives (e.g., *SLH-DSA* with a 7856-byte signature, nearly 3× that of *ML-DSA*), we mainly focus on *ML-DSA* for direct comparison. While *FN-DSA-1024*[57] offers smaller signatures and similar performance, it relies on floating-point operations, making it less suitable for mobile platforms (UE). Even with efficient implementation, *FN-DSA* still incurs fragmentation, albeit less than *ML-DSA*. Given *ML-DSA*'s relative simplicity and prominence in the NIST PQC process, we adopt it as the main PQ baseline. Threshold variants of all PQC schemes are expected to incur significantly higher overhead, rendering them impractical for our use case.

Given that *EC-Schnorr*'s structure is inherently more amenable to threshold signing, supports practical implementation optimizations (e.g., [71]), and exhibits comparable timings and sizes to *ECDSA* [43], we adopt *EC-Schnorr* [70] as the foundation for our scheme and the primary baseline for comparison. For conventionally secure signatures, we consider *EC-Schnorr* [43] as an optimized standard, *BLS* [69] for its aggregation capabilities, and *Schnorr-HIBS* [8] as a closely related certificateless scheme. Among closely related approaches, the scheme in [2] utilizes a certificate chain with three distinct signature schemes, where our performance evaluation covers their architecture using standardized signature metrics. The work in [4] introduces a "broadcast but verify" architecture using certificate-chain cryptography for 5G bootstrapping, proposing a separate *signingSIB* message instead of embedding signatures in SIBs to improve efficiency. Notably, BORG can serve as a drop-in replacement in their design, enhancing both efficiency and security. *BARON* [10], a token-based protocol using symmetric encryption enables UE authentication but does not protect SIBs, allowing tampering without token invalidation, and is thus excluded from direct comparison. All baselines are evaluated against both centralized and threshold variants of BORG.

Table 1
Quantitative comparison of the alternative signature schemes in an overly ideal scenario (*sending only signature*) for authenticating 5G cellular BSs.

Scheme	Sign (ms)	Verif. (ms)	Packet Proc.(ms)	Transmission (ms)	Crypto./Comm.(B)	PK (B)	E2E Delay (ms)
BLS [69]	0.42	1.15	0.03	< 0.01	48/–	96	1.60
ML-DSA [36]	0.12	0.03	0.57	160.02-1280.02	2420/2976	1312	160.74-1280.74
Schnorr-HIBS† [8]	0.30	1.27	0.04	< 0.01	64/–	32	1.61
Centralized-BORG	0.33	1.27	0.04	< 0.01	64/–	32	1.64
(2,3)-BORG*	1.12	1.27	0.04	< 0.01	64/–	32	2.43
(2,3)-BORG	1.68	1.27	0.04	< 0.01	64/–	32	2.99

E2E delay represents the total time for signature generation, 5G delay (packet processing and transmission), and signature verification. A dash (–) indicates *no additional* overhead, i.e., fits within the default SIB1 packet. $(2,3)$-BORG* considers a precomputed preprocessing phase. †*EC-Schnorr* [70] shares identical timing and size metrics with *Schnorr-HIBS* [8].

6.4. Experimental Results

Table 1 presents a quantitative comparison of candidate schemes for signing a single SIB1 message, evaluating signing/verification time, 5G processing, cryptographic/communication overhead, and end-to-end (E2E) latency. Table 2 extends this with both qualitative and quantitative analysis in the full 5G hierarchical bootstrapping context. We report the average results for 10000 iterations for all the schemes when not set up in the 5G testbed. As the 5G testbed requires manual intervention, we report the average of 10 iterations when the schemes are run on the testbed.

6.4.1. Quantitative Comparison

This section presents the computational and communication overhead of BORG, accompanied by a comparison to alternative signatures.

• *Computational Costs:* We begin by analyzing the signing and verification complexity for a single SIB1 message, with the results summarized in Table 1. *EC-Schnorr* and *BLS* exhibit low E2E delay, with *BLS* incurring slightly higher computational cost due to pairing-based operations. While *ML-DSA* achieves comparable execution times through optimized implementation, its large signature size results in substantial 5G communication overhead and a total delay of 1280.74 ms, making it impractical for 5G SIB1 authentication, even without considering certificate hierarchy. *SLH-DSA*, with a signature nearly three times larger and slower signing and verification (11 ms and 0.84 ms, respectively), is even less suited for 5G authentication. *Schnorr-HIBS* and *Centralized−BORG* exhibit nearly identical execution time and communication overhead, resulting in comparable E2E delay for signing a single SIB1 message. In the threshold BORG variant (e.g., $(2,3)$ configuration), signing takes approximately 1.12 ms without preprocessing and 1.68 ms with preprocessing included. Its verification time matches that of *Schnorr-HIBS* and *Centralized−BORG*, which is particularly crucial for the resource-constrained UEs. Also, BORG adopts Ringtail [44] as the ThPQ instantiation for distributed audit logging due to its performance benefits. Since ThPQ only signs the SIB1 signatures and does not affect BS/UE operations, it is excluded from the core performance comparison. Its preprocessing, signing, and verification take 89.4, 3.29, and 1.2 ms, respectively.

In the full 5G evaluation (Table 2), which accounts for transmission of keys, certificates, and IDs, BS signing costs remain consistent with those reported in Table 1. For full verification, however, the UE must validate the SIB1 signature and, in certificate-based schemes, also verify certificates for the AMF and BS. This highlights the efficiency advantage of hierarchical schemes over flat alternatives such as *BLS*, *EC-Schnorr*, and *ML-DSA*. While *BLS* benefits from signature aggregation, its pairing-based verification introduces considerable computational cost. Similarly, while *ML-DSA* offers relatively fast verification, its large key and signature sizes and high communication overhead result in a substantial E2E delay, rendering it infeasible for 5G bootstrapping, where SIB1 packets are sent every $20 \sim 160$ ms. Our *Centralized*-BORG achieves a total delay of 1.64 ms, making it approximately $404 \sim 3221\times$ faster than *ML-DSA* ($662.47 \sim 5282.47$ ms). The *(2,3)-BORG*, is slightly slower than the centralized version and *Schnorr-HIBS*, yet remains significantly faster ($222 \sim 1767\times$) than *ML-DSA*. Like *Schnorr-HIBS*, both centralized and threshold BORG variants transmit only 144 bytes of cryptographic artifacts, fitting entirely within a single SIB1 packet, demonstrating the superior efficiency of our future-proof scheme over existing NIST-PQ solutions.

• *Communication Overhead:* In 5G BS authentication, certificate-based schemes must transmit the SIB1 message, signature, public keys, and two certificates (for the AMF and BS) to establish key authenticity. In contrast, hierarchical schemes transmit only the SIB1 signature, corresponding public keys, and identities, eliminating certificates and reducing communication overhead. As shown in Table 2, flat schemes such as *BLS* and *EC-Schnorr* incur higher cryptographic overhead, while *ML-DSA* imposes substantial cryptographic and communication costs. Specifically, *ML-DSA* suffers from fragmentation, requiring 34 network packets and incurring a total overhead of 12276 bytes. In contrast, hierarchical schemes like *Schnorr-HIBS* and both centralized and threshold variants of BORG maintain a compact 144-byte overhead, fitting within a single SIB1 packet. While the threshold-BORG requires inter-gNB communication for signature aggregation via the XnAP interface [72] (using SCTP over IP [73]), this delay is implementation-dependent and typically below 10 ms [74]. Even under

Table 2
Comparison of candidate signature schemes for authenticating SIB1.

Scheme	System Architecture and Features	Sign Delay (ms)	Full Verification Delay (ms)	5G Delay (ms)	Crypto./Comm. Overhead (B)	E2E Delay (ms)
BLS [69]	2-Level Certificate with Aggregation	0.42	3.46	0.05	240/–	3.93
EC-Schnorr [70]	2-level Certificate	0.30	3.80	0.05	256/–	4.15
ML-DSA [36]	2-Level Certificate	0.12	0.12	662.23 ∼ 5282.23	9884/12276	662.47 ∼ 5282.47
Schnorr-HIBS [8]	Hierarchical	0.30	1.27	0.04	144/–	1.61
Centralized-BORG	Hierarchical, Fail-Stop	0.33	1.27	0.04	144/–	1.64
(2,3)-BORG	Hierarchical, Fail-Stop, Threshold	1.68	1.27	0.04	144/–	2.99

E2E delay presents the total time for signature generation, 5G delay and full verification. A dash (–) indicates *no (additional)* overhead, i.e., fits within the default SIB1 packet.

this upper bound, *(2,3)-BORG* remains significantly more efficient than NIST-PQ alternatives requiring fragmentation.

• *UE Side Overhead:* Due to the resource-constrained setting, it is imperative to investigate the computational overhead in the UE, posed by the schemes. We observe that all the schemes (including ours) except ML-DSA introduce only $0.015 - 0.02$ ms additional packet processing times in the UE. The potential reason for this is that UE only needs to process one SIB1 packet containing few additional bytes pertaining to the associated signature. In contrast, the packet processing time for ML-DSA on the UE side is much higher (0.26ms). However, note that UE still needs to verify the signature after processing the packet. For verification, we observe around $1.15 - 1.27$ms overhead for all the schemes (including ours) except ML-DSA. For verification, ML-DSA is faster with only 0.03ms overhead. However, note that the primary concern with ML-DSA is its much larger communication overhead and the complexity it introduces in the protocol stack. Moreover, in the event of packet loss, BORG remains robust, as the signature can be delivered in the next broadcast of the SIB1 message without causing any packet-loss-related communication overhead. In contrast, NIST-PQC, like ML-DSA, requires synchronization from multiple consecutive SIB1 as discussed in section III.

6.4.2. Qualitative Comparison

This section presents a qualitative comparison of BORG against related approaches, including certificate-based, hierarchical, and threshold schemes, in the context of 5G bootstrapping authentication.

• *Limitations of Authentication with Flat Hierarchy:* Direct use of non-hierarchical signatures for 5G BS authentication requires a certificate chain to authenticate public keys, increasing communication overhead and requiring the UE to verify both the SIB1 signature and the certificates for the AMF and BS keys. This adds considerable computational burden and is essential to mitigate threats like FBSs and MiTM attacks. As shown in Table 2, even efficient, conventionally secure schemes such as *EC-Schnorr* and *BLS* incur notable cryptographic overhead. For instance, if SIB1 configurations exceed 120 bytes, *EC-Schnorr* requires fragmentation and delivery over two packets. Full-PQC schemes are even more demanding, with total communication costs nearing 12 KB and requiring extensive fragmentation, posing serious reliability and availability issues. This makes PQC signatures not only computationally infeasible but also vulnerable to authentication failure if any signature fragment is lost in transit.

• *Limitations of Thresholding for 5G Authentication in the PQ Era:* Threshold signatures, particularly those offering PQ guarantees, impose substantial overhead, making them impractical for 5G. Even conventionally secure threshold schemes, such as Schnorr-based signatures (e.g., *FROST* [27]), which resemble *Schnorr-HIBS* and $(2, 3)$-BORG, incur higher computational costs when applied to SIB1 authentication in a $(2, 3)$ setting. Several threshold variants of NIST-PQC signatures rely on resource-intensive techniques: multi-party computation (e.g., threshold-*Dilithium* and threshold-*FN-DSA* reportedly require 12 s and 6 s to sign [75]), homomorphic hashing and commitments (e.g., *Dilizium* [76] incurs hundreds of milliseconds and 21120-byte signatures), and fully homomorphic encryption [77], which leads to delays in the order of seconds. Additional constructions, such as those based on the Fiat–Shamir with Aborts paradigm [78] or hash-and-sign lattice approaches, suffer from transformation complexity and abort management overhead. In contrast, BORG offers a lightweight, practical alternative for 5G BS authentication. It maintains distributed trust and compromise resilience through thresholding while supporting FS security and PQ forgery detection, achieving a strong balance between security and deployability.

• *PQ Assurances:* As detailed in Section 4, NIST-PQC signatures are currently unsuitable for 5G SIB1 authentication due to their large sizes, high computational costs, and substantial communication overhead. These signatures exceed the SIB packet size limit. Experimental results (Tables 1-2) show that full-PQC schemes require excessive fragmentation and processing delays; for instance, *ML-DSA* requires 8 additional packets for signature delivery, 12 with the public key, and 33 when including a 2-level certificate chain. Even without new message types, repeated SIB1 transmissions must carry fragmented signatures, further stressing the system. Broadcast unreliability and physical-layer constraints exacerbate deployment challenges. While conventional-secure schemes are efficient, they lack forgery

detection and offer no protection against compromised BSs. BORG achieves 3 orders of magnitude faster execution and 85× lower communication overhead than full-PQC alternatives like *ML-DSA*, while offering distributed trust, forgery detection, and breach resiliency. Its efficiency, comparable to conventional hierarchical schemes, makes it a practical solution for 5G BS authentication.

7. Related Work

We organize prior work on 5G BS authentication by technical approach, analyzing each category's contributions and limitations with respect to the design goals of BORG.

PKI-Based BS Authentication. Early proposals adapted PKI frameworks to protect SIB messages. Lee et al. [5] and Zheng [6] established certificate-based foundations for mobile network authentication. Hussain et al. [2] provided the first systematic analysis of 5G bootstrapping vulnerabilities and proposed attaching signatures and certificate chains to SIB1 and SIB2 messages. Ross et al. [4] proposed a "broadcast-but-verify" model that transmits the signature in a separate *signingSIB* message to decouple authentication overhead from SIB1. Gao et al. [17] explored delegated signing to reduce per-BS computational cost, and Wuthier et al. [79] combined multi-factor authentication with offline blockchain-based certificate delivery. The 3GPP standardization body has explored PKI-based SIB protection in TR 33.809 [7], though SIB1 remains unprotected in the current RRC specification [21]. All PKI-based schemes share a fundamental limitation: certificate chains for AMF and BS keys routinely exceed the 372-byte SIB1 limit, requiring fragmentation across multiple packets, compounding verification cost on resource-constrained UEs, and remaining entirely vulnerable to quantum-capable adversaries.

Token- and Symmetric-Based Schemes. TESLA [11] introduced lightweight broadcast authentication using a one-way key chain with timed key disclosure, relying solely on symmetric primitives at the cost of loose time synchronization. BARON [10] employed symmetric tokens for pre-authentication defense, introducing the concept of a Closed Trusted Entity (CTE) for secure connection initialization and handover in 5G networks. While these approaches avoid asymmetric certificate overhead, they do not protect SIB content itself: broadcast parameters remain susceptible to tampering without invalidating any token or key chain.

Identity-Based and Certificate-Free Schemes. To eliminate certificate chains, IBS-based schemes derive BS and AMF signing keys hierarchically from a master key pre-installed in the USIM, achieving compact overhead within a single SIB1 packet. Singla et al. [8] introduced *Schnorr-HIBS*, a hierarchical IBS scheme that serves as the closest conventional-secure baseline to BORG. Ramadan et al. [19] explored server-aided IBS to offload UE verification cost. Yu et al. [20] and Sun and Peng [9] further refined two-level HIBS constructions for LTE/5G. While IBS schemes achieve the best efficiency among conventional approaches, all rely on ECDLP hardness and are therefore broken by quantum-capable adversaries. Critically, none provides distributed trust, forgery detection, or any accountability mechanism following a BS compromise.

Threshold and Distributed Authentication. A small number of efforts have explored threshold signatures to distribute signing responsibility across multiple BSs. Sengupta and Lakshminarayanan [12] explored online-offline threshold IBS for 5G IoT settings, and Vikhrova et al. [13] examined multi-SIM support scenarios involving coordinated BS authentication. However, these efforts operate under idealized assumptions and do not address the combined requirements of efficiency under 5G packet-size constraints, accountability upon BS compromise, or long-term breach resiliency. The broader challenge of designing a threshold authentication scheme that simultaneously meets 5G's strict overhead constraints and provides verifiable forgery detection remains unaddressed in prior work.

Privacy and AKA-Layer Security. Orthogonal to BS broadcast authentication, a line of work addresses privacy and mutual authentication at the NAS layer. Alnashwan et al. [80] presented a UC-secure authentication and handover protocol with strong user privacy guarantees, and Wang et al. [81] proposed encryption and KEM-based countermeasures targeting linkability vulnerabilities in 5G-AKA. These protocols operate after the bootstrapping phase and are orthogonal to SIB broadcast authentication; they assume a trustworthy BS connection has already been established, precisely the guarantee that BORGis designed to provide.

Post-Quantum and Hybrid Solutions. Recent work has begun addressing long-term quantum threats in cellular security. Efforts on PQ-AKA protocols [52, 53, 82] apply lattice-based KEMs and PQ identification schemes to the NAS-layer authentication and key agreement procedures. Hybrid constructions combining NIST-PQC KEMs with symmetric primitives [14, 15, 16] aim to reduce the overhead of full PQC adoption for primary UE authentication. However, all of these efforts target unicast session establishment and are structurally incompatible with the one-to-many SIB broadcast authentication setting: KEMs establish shared secrets between two parties, and applying full PQC signatures to SIBs would compound fragmentation rather than resolve it, as demonstrated in Section 4. None of these works addresses the unique constraints of initial BS bootstrapping authentication under 5G's strict size, timing, and broadcast requirements.

8. Limitations, Conclusion, and Future Work

In conclusion, our feasibility assessment exposes the practical limitations of directly integrating NIST-PQC and conventional signature schemes into 5G bootstrapping authentication, primarily due to excessive signature sizes, certificate overhead, and fragmentation. To future-proof the 5G authentication, we propose BORG, a lightweight, distributed, and compromise-resilient framework that enables threshold authentication with forgery detection while meeting strict 5G constraints. Although BORG does not provide full PQ security, it is designed to operate across all bootstrapping connections and currently secures critical SIB messages.

It is also worth noting that BORG does not address privacy concerns related to UE-to-BS connections, passive eavesdropping, or other physical-layer threats, and remains vulnerable to overshadow attacks. Future work will extend protection to mitigate overshadow attacks in the PQ era. Looking ahead to 6G, we expect BORG to help accelerate the adoption of secure and practical bootstrapping mechanisms beyond what 5G was able to incorporate.

Acknowledgment

This work is supported by the CNS (2350213), NSF Grant No. 2112471, the University of Texas System Rising STARs Award (No. 40071109), and the startup funding from the University of Texas at Dallas.

A. APPENDIX

A.1. Acronyms

A complete list of acronyms used throughout the paper is provided in Table 3.

A.2. Signature Verification Correctness

We demonstrate the correctness of the verification procedure by proving that an honestly generated signature in BORG signature scheme satisfies the verification equation.

Let $\sigma_{k,j} = (R_j, z_j)$ be a valid signature at hierarchy level k, generated according to the BORG.Sign algorithm. Any verifier can validate this signature using the BORG.MVerify procedure (Algorithm 3), by computing the intermediate values h_{ID_ℓ} for each $\ell = 1, 2, \ldots, k$, as well as Q and h_j. The correctness of the signature is verified by checking whether the following equation holds:

$$g^{z_j} \stackrel{?}{=} R_j \cdot (Q \cdot Q_{ID_k} \cdot (PK_{ID_0})^{\prod_{\ell=1}^{k} h_{ID_\ell}})^{h_j} \mod p$$

which confirms the integrity and authenticity of the signed message under the Schnorr-based structure of the BORG scheme. By Lagrange interpolation, the secret shares satisfy:

$$\sum_{i=1}^{\beta} \lambda_i \cdot sk_{ID_{k,i}} = sk_{ID_k}.$$

Thus, if all signers behave honestly, then:

$$z_j = \sum_{i=1}^{\beta} z_{i,j} = \sum_{i=1}^{\beta} (d_{i,j} + e_{i,j} \cdot \rho_{i,j}) + h_j \cdot \sum_{i=1}^{\beta} \lambda_i \cdot sk_{ID_{k,i}} \mod q.$$

Hence, the left-hand side of the verification algorithm is:

$$g^{z_j} = g^{r_j} \cdot g^{h_j \cdot sk_{ID_k}} = R_j \cdot g^{h_j \cdot sk_{ID_k}} \mod p.$$

Given the hierarchical key extraction procedure ensures:

$$g^{sk_{ID_k}} = Q \cdot Q_{ID_k} \cdot \left(PK_{ID_0}\right)^{\prod_{\ell=1}^{k} h_{ID_\ell}} \mod p,$$

where:

$$Q \leftarrow \prod_{\ell=1}^{k-1} (Q_{ID_\ell})^{\prod_{\omega=\ell+1}^{k} h_{ID_\omega}}$$

Putting it together:

$$g^{z_j} = R_j \cdot \left(Q \cdot Q_{ID_k} \cdot PK_{ID_0}^{\prod_{\ell=1}^{k} h_{ID_\ell}}\right)^{h_j} \mod p.$$

This matches the verifier's equation. Therefore, the verification algorithm accepts the signature.

Table 3
List of Acronyms.

Acronyms	Description
3GPP	Third Generation Partnership Project
5G-RAN	5G Radio Access Network
5G-CN	5G Core Network
AMF	Access and Mobility Management Function
AKA	Authentication and Key Agreement
BS	Base Station
CKG	Core Key Generator
DL-SCH	Downlink Shared Channel
ECDLP	Elliptic Curve Discrete Logarithm Problem
EUF-sID-CMIA	Existential Unforgeability under a Selective-ID adaptive Chosen Message-and-ID Attacks
FBS	Fake Base Station
FS	Fail-Stop
gNB	Next Generation NodeB
HIBS	Hierarchical Identity-Based Signature
IBS	Identity-Based Signature
ID	Identity of an entity (e.g., MAC address)
IMSI	International Mobile Subscriber Identity
LTE	Long Term Evolution
MAC	Message Authentication Code
MIB	Master Information Block
MiTM	Man-in-The-Middle
MPK	Master Public Key
msk	Master Secret Key
NAS	Non Access Stratum
NIST	National Institute of Standards and Technology
PK	Public Key
PKI	Public Key Infrastructure
PLMN	Public Land Mobile Network
PM	Post-Mortem
PQC	Post-Quantum Cryptography
RRC	Radio Resource Control
SDR	Software Defined Radio
SIB	System Information Block
sk	Secret Key
UE	User Equipment
USIM	Universal Subscriber Identity Module

A.3. Security Proof

Theorem 1. *If an adversary $\mathcal{A}$ can $(q_E, q_P, q_S, q_{H_1,H_2})$-break BORG in the random oracle model (Definition 3.1) with an advantage ϵ in time τ while having access to at most $(t-1)$-out-of-n signing participants, where q_E, q_P, q_S, q_{H_1}, and q_{H_2} denote queries to key extraction, preprocessing, signing, and hash functions H_1 and H_2, then an algorithm C can be constructed to break the (EC)DLP in group $\mathbb{G}$.*

Proof. We assume that the adversary $\mathcal{A}$ is able to compromise $t-1$ signing participants by accessing the key extraction oracle $\mathcal{O}_E$. It can also query the preprocessing oracle $\mathcal{O}_P$, signing oracle $\mathcal{O}_S$, and hash function oracles $\mathcal{O}_{H_1,H_2}$. We assume there are t users in each level of the hierarchy. We note that the security proof can be generalized to n users with t-out-of-n thresholding. $\mathcal{A}$ has control over $(t-1)$ signing participants. We consider a challenger C which invokes

$\mathcal{A}$ as a black box and handles input and output queries, simulating the honest signing participant (P_t) across all queries and algorithms. By embedding a random challenge (in this case, a DLP instance) $\omega = g^{a^*} \in \mathbb{G}$ in query responses for the target ID^* chosen by the forger. $\mathcal{A}$ starts by picking a target identity $\vec{ID}^* = (ID_1^*, \dots, ID_\ell^*) \in \mathbb{Z}_p^\ell$ as the challenge identity.

• **Setup:** Given κ, $\mathcal{C}$ executes the BORG.Setup(1^κ) (Algorithm 1). It randomly selects $\alpha_0 \xleftarrow{\$} \mathbb{Z}_q$, derives $sk_{ID_0} \leftarrow H_1(\alpha_0)$, and computes $PK_{ID_0} \leftarrow g^{sk_{ID_0}} \mod p$. It then provides $(PK_0, params)$ to $\mathcal{A}$ while keeping (α_0, sk_{ID_0}) secret.

• **Execute** $\mathcal{A}^{\mathcal{O}_E, \mathcal{O}_P, \mathcal{O}_S}(PK_{ID_0}, params)$**:** $\mathcal{A}$ can adaptively issue a polynomially bounded number of queries, with $\mathcal{C}$ acting as the honest party P_t as follows.

- *Secret Key Extraction Oracle ($\mathcal{O}_E$):* For a query identity $\vec{ID} = (ID_1, \dots, ID_\ell) \in \mathbb{Z}_p^\ell$, where $\vec{ID} \neq \vec{ID}^*$ or any $ID_i^* \notin \vec{ID}$ for $i = 1, \dots, \ell$, $\mathcal{C}$ follows the BORG.Extract(.) procedure and returns $(\{sk_{ID_{\ell,i}}\}_{i=1}^t, \{PK_{ID_{\ell,i}}\}_{i=1}^t, \vec{Q}_{ID_\ell})$. For $\vec{ID} = \vec{ID}^*$, $\mathcal{C}$ embeds the challenge $\omega \leftarrow g^{a^*} \mod p$ by setting $PK_{ID_{\ell,t}} = \omega$. It then derives the secret and public keys for the remaining $t-1$ participants by choosing $a_i \xleftarrow{\$} \mathbb{Z}_q$ and computing $PK_{ID_{\ell,i}} \leftarrow \omega^{\lambda_t} \cdot g^{\sum_{i=1}^{t-1} \lambda_i \cdot a_{\ell,i}}$ for $i = 1, \dots, t-1$. The group verification key Q_{ID^*} and $\vec{Q}_{ID^*}$, follows the same procedure as the BORG.Extract(.).
- *Preprocessing Oracle ($\mathcal{O}_P$):* Given J and the ID of the signing participants (where $\vec{ID} \neq \vec{ID}^*$ or any $ID_i^* \notin \vec{ID}$ for $i = 1, \dots, \ell$), it returns the commitment value of the participants $(\mathcal{L}_i \leftarrow (i, \{E_{i,j}\}_{j=1}^J, \{D_{i,j}\}_{j=1}^J))$ for $i = 1, \dots, t-1$, following BORG.Preprocess(.).
- *Signing Oracle ($\mathcal{O}_S$):* For a message m and $\vec{ID} = (ID_1, \dots, ID_\ell) \in \mathbb{Z}_p^\ell$, where $\vec{ID} \neq \vec{ID}^*$ or any $ID_i^* \notin \vec{ID}$ for $i = 1, \dots, \ell$, responds by using the BORG.Extract(.) algorithm to extract the secret key, obtain the commitment values BORG.Preprocess(.), and produce a valid signature relying on the signing algorithm BORG.Sign(.). The signature queries on the target ID^* would be rejected.
- *Hashing with a Random Oracle ($\mathcal{O}_{H_1}, \mathcal{O}_{H_2}$):* Let $\mathcal{L}_{H_1}$ and $\mathcal{L}_{H_2}$ denote the query logs for the hash functions H_1 and H_2, respectively. Upon receiving a query to H_1 on the tuple $(ID_\ell, \vec{Q}_{ID_\ell})$, the challenger $\mathcal{C}$ checks $\mathcal{L}_{H_1}$: if an entry exists, it returns the stored value; otherwise, it samples $x \xleftarrow{\$} \mathbb{Z}_q$, stores it in $\mathcal{L}_{H_1}$, and returns x. Similarly, for a query to H_2 on (R_j, Q_{ID_ℓ}, m_j), $\mathcal{C}$ checks $\mathcal{L}_{H_2}$ and either returns the stored value or samples $x' \xleftarrow{\$} \mathbb{Z}_q$, stores it, and returns x'.

• ***Forgery of $\mathcal{A}$:*** $\mathcal{A}$ produces a valid signature (m^*, σ^*) for $\vec{ID}^*$ under the master and group public keys (PK_{ID_0}, Q_{ID_k}). $\mathcal{A}$ wins the experiment if satisfies the conditions mentioned in Definition 3.1.

• ***Solution to DLP via $\mathcal{C}(\omega)$:*** Given the adversary $\mathcal{A}$ with access to $t-1$ signing participants can produce a signature forgery σ^*, $\mathcal{C}$ can uses $\mathcal{A}$ as a black-box forger, and utilize the generalized forking lemma, solves the DLP for the embedded challenge ω, as shown below:

- Applying GFL, the adversary $\mathcal{A}$ is run twice with the same random tape while having different hash queries to $\mathcal{O}_{H_2}$, then, $\mathcal{C}$ can obtain two signature forgeries $\sigma^* = ({R_j}^*, {z_j}^*), \sigma^{*\prime} = ({R_j}^*, {z_j}^{*\prime})$.
- Given the query responses from preprocessing and $\mathcal{H}_1$ are the same, we get to the following two equations:

$$\sum_{i=1}^{t} {z^*}_{i,j} = \sum_{i=1}^{t} {d^*}_{i,j} + \sum_{i=1}^{t} {e^*}_{i,j} \cdot {\rho^*}_{i,j} + \sum_{i=1}^{t} \lambda_i \cdot sk_{ID_{i,j}} . {h^*}_j \mod q$$

$$\sum_{i=1}^{t} {z^{*\prime}}_{i,j} = \sum_{i=1}^{t} {d^*}_{i,j} + \sum_{i=1}^{t} {e^*}_{i,j} \cdot {\rho^*}_{i,j} + \sum_{i=1}^{t} \lambda_i \cdot sk_{ID_{i,j}} . {h^{*\prime}}_j \mod q$$

- From the above equations, we get:

$$sk_{ID_\ell} = \frac{\sum_{i=1}^{t} (z^*_{i,j} - z^{*\prime}_{i,j})}{h^*_j - h^{*\prime}_j}$$

$$a^* = \frac{1}{\lambda_t} \times \left(\frac{\sum_{i=1}^{t} (z^*_{i,j} - z^{*\prime}_{i,j})}{h^*_j - h^{*\prime}_j} - \sum_{i=1}^{t-1} \lambda_i \cdot sk_{ID_\ell} \right)$$

- Finally, $\mathcal{C}$ obtains the DLP of ω in $\mathbb{G}$.

□

Theorem 2. *BORG provides λ_1-bit signer-side fail-stop security against quantum-capable adversaries controlling up to $(t-1)$-out-of-n signing participants, as formalized in Definition 3.2, and λ_2-bit non-repudiation security against quantum adversaries with access to one-out-oft signing participants, as captured in Definition 3.3. Both guarantees rely on the hardness of breaking the second preimage resistance of a cryptographically secure hash function, while providing κ bit verifier-side security via EUF-sID-CMIA in the random oracle model as defined in Definition 3.1.*

Proof. Let $\mathcal{A}$ be a quantum-capable adversary with access to the signing oracle, preprocessing oracle, and control over $(t-1)$ out of n signing participants. We assume t users per level in the hierarchy, though the proof generalizes to t-out-of-n thresholding. Assume, for contradiction, that $\mathcal{A}$ succeeds in the signer-side fail-stop experiment (Definition 3.2) with non-negligible advantage ϵ, producing a forged signature σ_k^* on a message m^* that (i) passes verification ($1 \leftarrow$ BORG.MVerify$(m^*, \vec{ID}_k, \vec{Q}_{ID_k}, \sigma_k^*)$), and (ii) cannot be proven invalid via $0 \leftarrow$ BORG.PoF$(\{\hat{e}^*_{i,j}\}_{i=1}^t, \{\hat{d}^*_{i,j}\}_{i=1}^t, m^*, \sigma_k^*, hist)$, i.e., honest signers fail to produce a valid forgery proof. We construct a reduction algorithm $\mathcal{C}_1$ that uses $\mathcal{A}$ as a black box, handles queries, and simulates the honest signing participant (P_t) across all queries and algorithms. For the target message m^*, $\mathcal{C}_1$ embeds a second preimage challenge

as a commitment from an honest signer by programming commitment values $(e^*_{i,t} \leftarrow H_1(\hat{e}^*_{t,j}||j||ID_{k,t}), d^*_{i,t} \leftarrow H_1(\hat{d}^*_{t,j}||j||ID_{k,t}))$ where the random nonces are $(\hat{e}^*_{i,t}, \hat{d}^*_{i,t}) \xleftarrow{\$} \mathbb{Z}_q \times \mathbb{Z}_q$.

Since forgery detection in BORG.PoF depends on these hash-based commitments that derive the shared component R_j, any successful forgery must reproduce these commitments without access to the original random nonces. Thus, if $\mathcal{A}$ outputs a successful forgery σ^* and corresponding alternate preimage $(\hat{e}^{*\prime}_{i,t}, \hat{d}^{*\prime}_{i,t})$ that satisfy the conditions mentioned in Definition 3.2 such that $H_1(\hat{e}^{*\prime}_{i,t}||j||ID_{k,t}) = H_1(\hat{e}^*_{i,t}||j||ID_{k,t})$ and $H_1(\hat{d}^{*\prime}_{i,t}||j||ID_{k,t}) = H_1(\hat{d}^*_{i,t}||j||ID_{k,t})$ while $(\hat{e}^{*\prime}_{i,t}, \hat{d}^{*\prime}_{i,t}) \neq (\hat{e}^*_{i,t}, \hat{d}^{*\prime}_{i,t})$, then $\mathcal{C}_1$ outputs $(\hat{e}^{*\prime}_{i,t}, \hat{d}^{*\prime}_{i,t})$ as a valid second preimage for the embedded challenge. This contradicts the assumed hardness of second preimage resistance of H_1. Hence, under Grover's algorithm [83], the quantum adversary's success probability is reduced to $\mathcal{O}(2^n)$, yielding λ_1-bit PQ security for an n-bit hash function. □

Based on Definition 3.3, we prove fail-stop non-repudiation of BORG under the hardness of second preimage resistance of a cryptographically secure hash function.

Proof. Let $\mathcal{A}$ be a quantum-capable adversary with access to preprocessing queries and control over one of the t signing participants. Suppose $\mathcal{A}$ wins the non-repudiation experiment in Definition 3.3 with non-negligible probability ϵ; that is, $\mathcal{A}$ participates in generating a valid signature σ_k such that (i) it passes verification $1 \leftarrow$ BORG.MVerify$(m, \vec{ID}_k, \vec{Q}_{ID_k}, \sigma_k)$, and (ii) later constructs a forged proof π^* satisfying $1 \leftarrow$ BORG.PoFVerify$(\alpha_k, sk_{ID_{k-1}}, \vec{Q}_{ID_k}, m, \sigma'_k, \pi^*)$, despite σ_k being honestly generated according to the signing protocol. We construct a reduction algorithm $\mathcal{C}_2$ that treats $\mathcal{A}$ as a black box, simulates all t signers, and handles all input/output queries. For the message m, $\mathcal{C}_2$ embeds second preimage challenges as the commitments of all t signers by programming values $(e^*_{i,j} \leftarrow H_1(\hat{e}^*_{i,j}||j||ID_{k,i}), d^*_{i,j} \leftarrow H_1(\hat{d}^*_{i,j}||j||ID_{k,i})$ where $(\hat{e}^*_{i,j}, \hat{d}^*_{i,j}) \xleftarrow{\$} \mathbb{Z}_q \times \mathbb{Z}_q$ for all $i = 1, \ldots, t$.

Since forgery detection in BORG.PoF relies on hash-based commitments used to derive the shared component R_j, a valid forgery must reproduce these commitments without access to the original nonces. For the t signers to falsely prove a legitimate signature σ_k as a forgery, at least one signer must find a distinct preimage $\hat{e}^{*\prime}_{i,j} \neq \hat{e}^*_{i,j}$ such that $H_1(\hat{e}^{*\prime}_{i,j}||j||ID_{k,i}) = H_1(\hat{e}^*_{i,j}||j||ID_{k,i})$, for some $i \in \{1, \ldots, t\}$. If the quantum-capable adversary $\mathcal{A}$ outputs a valid-looking proof of forgery π^* using such alternate preimage $(\hat{e}^{*\prime}_{i,j}, \hat{d}^{*\prime}_{i,j})$ that satisfies the conditions mentioned in Definition 3.3, then the reduction $\mathcal{C}_2$ extracts a second preimage for the embedded challenge, contradicting the assumed hardness of second preimage resistance of H_1. Under Grover's algorithm [83], the adversary's success probability reduces to $\mathcal{O}(2^{n/2})$, yielding λ_2-bit PQ security for an n-bit hash function. □

References

[1] S. Wuthier, J. Kim, J. Kim, S.-Y. Chang, Fake base station detection and blacklisting, in: 2024 33rd International Conference on Computer Communications and Networks (ICCCN), IEEE, 2024, pp. 1–9.

[2] S. R. Hussain, M. Echeverria, A. Singla, O. Chowdhury, E. Bertino, Insecure connection bootstrapping in cellular networks: the root of all evil, in: Proceedings of the 12th conference on security and privacy in wireless and mobile networks, 2019, pp. 1–11.

[3] J. Cao, M. Ma, H. Li, R. Ma, Y. Sun, P. Yu, L. Xiong, A survey on security aspects for 3gpp 5g networks, IEEE communications surveys & tutorials 22 (1) (2019) 170–195.

[4] A. J. Ross, B. Reaves, Y. Nasser, G. Cukierman, R. P. Jover, Fixing insecure cellular system information broadcasts for good, in: Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses, 2024, pp. 693–708.

[5] C.-C. Lee, I.-E. Liao, M.-S. Hwang, An extended certificate-based authentication and security protocol for mobile networks, Information Technology and Control 38 (1) (2009).

[6] Y. Zheng, An authentication and security protocol for mobile computing, in: Mobile Communications: Technology, tools, applications, authentication and security IFIP World Conference on Mobile Communications Sep. 1996, Canberra, Australia, Springer, 1996, pp. 249–257.

[7] 3GPP TS 33.809 Study on 5G security enhancements against False Base Stations (FBS): Certificate based solution for Protecting System Information Messages with Digital Signature in an NPN., https://www.3gpp.org/ftp/TSG_SA/WG3_Security/TSGS3_100Bis-e/Docs/S3-202717.zip.

[8] A. Singla, R. Behnia, S. R. Hussain, A. Yavuz, E. Bertino, Look before you leap: Secure connection bootstrapping for 5g networks to defend against fake base-stations, in: Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security, 2021.

[9] Z. Sun, C. Peng, 5g-hcls: An authentication protocol to protect bootstrapping messages in 5g network, in: 2025 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2025.

[10] A. Lotto, V. Singh, B. Ramasubramanian, A. Brighente, M. Conti, R. Poovendran, Baron: Base-station authentication through core network for mobility management in 5g networks, in: Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks, 2023, pp. 133–144.

[11] A. Perrig, J. D. Tygar, A. Perrig, J. Tygar, Tesla broadcast authentication, Secure Broadcast Communication: In Wired and Wireless Networks (2003) 29–53.

[12] B. Sengupta, A. Lakshminarayanan, Fast verification of online/offline threshold signatures for 5g iot, in: 2024 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), IEEE, 2024, pp. 1–6.

[13] O. Vikhrova, S. Pizzi, A. Terzani, L. Araujo, A. Orsino, G. Araniti, Multi-sim support in 5g evolution: Challenges and opportunities, IEEE Communications Standards Magazine 6 (2) (2022) 64–70.

[14] R. C. Vuppala, D. Kumar, D. Je, N. Sharma, A. Nigam, D. Kim, Post-quantum secure hybrid methods for ue primary authentication in 6g with forward secrecy, in: GLOBECOM 2023-2023 IEEE Global Communications Conference, IEEE, 2023, pp. 2590–2595.

[15] Y. Ko, I. Pawana, I. You, 5g-aka-hpqc: Hybrid post-quantum cryptography protocol for quantum-resilient 5g primary authentication with forward secrecy, arXiv preprint arXiv:2502.02851 (2025).

[16] P. Scalise, R. Garcia, M. Boeding, M. Hempel, H. Sharif, An applied analysis of securing 5g/6g core networks with post-quantum key encapsulation methods, Electronics 13 (21) (2024) 4258.

[17] H. Gao, Y. Zhang, T. Wan, J. Zhang, On evaluating delegated digital signing of broadcasting messages in 5g, in: 2021 IEEE global communications conference (GLOBECOM), IEEE, 2021, pp. 1–7.

[18] Z. G. Al-Mekhlafi, S. A. Alfhaid, Innovative security measures: A comprehensive framework for safeguarding the internet of things, in: AI-Driven: Social Media Analytics and Cybersecurity, Springer, 2025.

[19] M. Ramadan, Y. Liao, F. Li, S. Zhou, Identity-based signature with server-aided verification scheme for 5g mobile systems, IEEE Access 8 (2020) 51810–51820.

[20] C. Yu, S. Chen, Q. Xing, Protecting unauthenticated messages in lte/5g mobile networks: A two-level hierarchical identity-based signature (hibs) solution, Computer Networks 254 (2024) 110814.

[21] 3GPP RRC Specification, https://www.etsi.org/deliver/etsi_ts/138300_138399/138331/18.01.00_60/ts_138331v180100p.pdf (2024).

[22] M. Polese, L. Bonati, S. D'oro, S. Basagni, Understanding o-ran: Architecture, interfaces, algorithms, security, and research challenges, IEEE Communications Surveys & Tutorials 25 (2) (2023).

[23] Z. G. Al-Mekhlafi, Software-defined vehicular networks (sdvn), International journal of computer science and network security: IJCSNS 22 (9) (2022) 231–243.

[24] D. Rupprecht, K. Kohls, T. Holz, C. Pöpper, Breaking lte on layer two, in: 2019 IEEE Symposium on Security and Privacy (SP), IEEE, 2019, pp. 1121–1136.

[25] L. Janzen, L. Becker, C. Wiesenäcker, M. Hollick, Oh no, my {RAN}! breaking into an {O-RAN} 5g indoor base station, in: 18th USENIX WOOT Conference on Offensive Technologies (WOOT 24), 2024, pp. 101–115.

[26] Interim Findings on KT Network Intrusion Event, https://www.msit.go.kr/eng/bbs/view.do;jsessionid=n9t3MMqrxQzovx5ap-ggsjQFttFLvDb326auIlJB.AP_msit_2?sCode=eng&mPid=2&mId=4&bbsSeqNo=42&nttSeqNo=1189.

[27] C. Komlo, I. Goldberg, Frost: flexible round-optimized schnorr threshold signatures, in: Selected Areas in Cryptography: 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers 27, Springer, 2021, pp. 34–65.

[28] L. De Simone, M. Di Mauro, R. Natella, F. Postiglione, Performance and availability challenges in designing resilient 5g architectures, IEEE Transactions on Network and Service Management (2024).

[29] S. Darzi, A. A. Yavuz, Counter denial of service for next-generation networks within the artificial intelligence and post-quantum era, arXiv preprint arXiv:2408.04725 (2024).

[30] Cost of a Data Breach Report 2025, https://www.ibm.com/reports/data-breach.

[31] 2025 Data Breach Investigations Report, https://www.verizon.com/business/resources/reports/dbir/.

[32] Inside the SK Telecom Data Breach: What Happened and What Companies Can Learn, https://www.alstonprivacy.com/inside-the-sk-telecom-data-breach-what-happened-and-what-companies.

[33] S. Darzi, K. Ahmadi, S. Aghapour, A. A. Yavuz, M. M. Kermani, Envisioning the future of cyber security in post-quantum era: A survey on pq standardization, applications, challenges and opportunities, arXiv preprint arXiv:2310.12037 (2023).

[34] ETSI TS 104 015 v1.1.1, Cyber Security (CYBER); Quantum-Safe Cryptography (QSC); Efficient Quantum-Safe Hybrid Key Exchanges with Hidden Access Policies, https://www.etsi.org/deliver/etsi_ts/104000_104099/104015/01.01.01_60/ts_104015v010101p.pdf.

[35] D. Sikeridis, P. Kampanakis, M. Devetsikiotis, Post-quantum authentication in TLS 1.3: A performance study, in: 27th Annual Network and Distributed System Security Symposium, NDSS 2020, San Diego, California, USA, February 23-26, 2020, The Internet Society, 2020.

[36] T. Dang, J. Lichtinger, Y.-K. Liu, C. Miller, D. Moody, R. Peralta, R. Perlner, Module-lattice-based digital signature standard, National Institute of Standards and Technology (NIST), Thinh Dang, Jacob (2024).

[37] H. Fourati, R. Maaloul, L. Chaari, M. Jmaiel, Comprehensive survey on self-organizing cellular network approaches applied to 5g networks, Computer Networks 199 (2021) 108435.

[38] S. S. Chow, L. C. Hui, S. M. Yiu, K.-P. Chow, Secure hierarchical identity based signature and its application, in: Information and Communications Security: 6th International Conference, ICICS 2004, Spain, October 27-29, 2004. Proceedings 6, Springer, 2004, pp. 480–494.

[39] D. Galindo, F. D. Garcia, A schnorr-like lightweight identity-based signature scheme, in: Progress in Cryptology: AFRICACRYPT 2009: Second International Conference on Cryptology in Africa, Gammarth, Tunisia, June, Proceedings 2, Springer, 2009, pp. 135–148.

[40] S. Ergezer, H. Kinkelin, F. Rezabek, A survey on threshold signature schemes, Network 49 (2020).

[41] W. Susilo, R. Safavi-Naini, J. Pieprzyk, Fail-stop threshold signature schemes based on elliptic curves, in: Australasian Conference on Information Security and Privacy, Springer, 1999, pp. 103–116.

[42] M. Yaksetig, Extremely simple (almost) fail-stop ecdsa signatures, Cryptology ePrint Archive (2024).

[43] D. Johnson, A. Menezes, S. Vanstone, The elliptic curve digital signature algorithm (ecdsa), International journal of information security 1 (2001) 36–63.

[44] C. Boschini, D. Kaviani, R. W. Lai, G. Malavolta, A. Takahashi, M. Tibouchi, Ringtail: Practical two-round threshold signatures from learning with errors, Cryptology ePrint Archive (2024).

[45] H. Kim, J. Lee, E. Lee, Y. Kim, Touching the untouchables: Dynamic security analysis of the lte control plane, in: 2019 IEEE Symposium on Security and Privacy (SP), 2019, pp. 1153–1168. doi:10.1109/SP.2019.00038.

[46] K. S. Mubasshir, I. Karim, E. Bertino, Gotta detect 'em all: Fake base station and multi-step attack detection in cellular networks (2025). arXiv:UsenixSecurity.

[47] N. Bennett, W. Zhu, B. Simon, R. Kennedy, W. Enck, Ransacked: A domain-informed approach for fuzzing lte and 5g ran-core interfaces, in: Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, CCS '24, Association for Computing Machinery, New York, NY, USA, 2024, p. 2027–2041. doi:10.1145/3658644.3670320.

[48] Safeguarding telecom networks against advance threats with ericsson's cyber defense solutions, https://www.ericsson.com/en/blog/2025/1/safeguarding-telecom-networks-with-ericssons-defense-solutions, accessed: October, 2025 (2025).

[49] C. J. Mitchell, The impact of quantum computing on real-world security: A 5g case study, Computers & Security 93 (2020).

[50] B. Pfitzmann, Fail-stop signatures: Principles and applications, in: Proc. Compsec, Vol. 91, Citeseer, 1991, pp. 125–134.

[51] C. Boschini, H. Dahari, M. Naor, E. Ronen, That's not my signature! fail-stop signatures for a post-quantum world, in: Annual International Cryptology Conference, Springer, 2024, pp. 107–140.

[52] G. Rossi Figlarz, F. Passuelo Hessel, Enhancing the 5g-aka protocol with post-quantum digital signature method, in: Intl. Conf. on Advanced Information Networking and Applications, Springer, 2024, pp. 99–110.

[53] M. T. Damir, T. Meskanen, S. Ramezanian, V. Niemi, A beyond-5g authentication and key agreement protocol, in: International Conference on Network and System Security, Springer, 2022, pp. 249–264.

[54] C.-P. Schnorr, Efficient identification and signatures for smart cards, in: Advances in Cryptology: CRYPTO'89 Proceedings 9, Springer, 1990.

[55] R. Safavi-Naini, W. Susilo, Threshold fail-stop signature schemes based on discrete logarithm and factorization, in: International Workshop on Information Security, Springer, 2000, pp. 292–307.

[56] J. J.-R. Chen, Y.-Y. Chiang, W.-H. Hsu, W.-Y. Lin, Fail-stop group signature scheme, Security and Communication Networks (1) (2021).

[57] D. Soni, K. Basu, M. Nabeel, N. Aaraj, M. Manzano, R. Karri, Falcon, Hardware Architectures for Post-Quantum Digital Signature Schemes (2021) 31–41.

[58] D. Cooper, et al., Stateless hash-based digital signature standard (2024).

[59] A. Shamir, How to share a secret, Communications of the ACM 22 (11) (1979) 612–613.

[60] M. Bellare, G. Neven, Multi-signatures in the plain public-key model and a general forking lemma, in: Proceedings of the 13th ACM

conference on Computer and communications security, 2006.
[61] A. Boldyreva, A. Palacio, B. Warinschi, Secure proxy signature schemes for delegation of signing rights, Journal of Cryptology 25 (2012) 57–115.
[62] ETSI TS 138 423 V15.8.0, 5G, NG-RAN, Xn Application Protocol (XnAP), https://www.etsi.org/deliver/etsi_ts/138400_138499/138423/15.08.00_60/ts_138423v150800p.pdf.
[63] A. A. Yavuz, System and method for secure review of audit logs, uS Patent 10,318,754 (Jun. 11 2019).
[64] S. E. Nouma, A. A. Yavuz, Practical cryptographic forensic tools for lightweight internet of things and cold storage systems, in: Proceedings of the 8th ACM/IEEE Conference on Internet of Things Design and Implementation, 2023, pp. 340–353.
[65] T. Le, P. Huang, A. A. Yavuz, E. Shi, T. Hoang, Efficient dynamic proof of retrievability for cold storage, Cryptology ePrint Archive (2022).
[66] C. Wendt, M. Barnes, Rfc 8588: Personal assertion token (passport) extension for signature-based handling of asserted information using tokens (shaken) (2019).
[67] Network Signal Guru User Manual, https://m.qtrun.com/docs/NSG_Manual_Aug_2017.pdf.
[68] G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, Y.-K. Liu, Status report on the third round of the nist post-quantum cryptography standardization process (2022).
[69] D. Boneh, B. Lynn, H. Shacham, Short signatures from the weil pairing, in: International conference on the theory and application of cryptology and information security, Springer, 2001, pp. 514–532.
[70] C.-P. Schnorr, Efficient signature generation by smart cards, Journal of cryptology 4 (1991) 161–174.
[71] G. Maxwell, A. Poelstra, Y. Seurin, P. Wuille, Simple schnorr multi-signatures with applications to bitcoin, Designs, Codes and Cryptography 87 (9) (2019) 2139–2164.
[72] ETSI TS 138 423 V16.2.0, 5G; NG-RAN; Xn Application Protocol (XnAP), https://www.etsi.org/deliver/etsi_ts/138400_138499/138423/16.02.00_60/ts_138423v160200p.pdf.
[73] ETSI TS 138 422 V17.0.0, 5G; NG-RAN; Xn signalling transport, https://www.etsi.org/deliver/etsi_ts/138400_138499/138422/17.00.00_60/ts_138422v170000p.pdf.
[74] R. Stewart, Stream control transmission protocol, Tech. rep. (2007).
[75] D. Cozzo, N. P. Smart, Sharing the luov: threshold post-quantum signatures, in: IMA International Conference on Cryptography and Coding, Springer, 2019, pp. 128–153.
[76] P. Laud, N. Snetkov, J. Vakarjuk, Dilizium 2.0: Revisiting two-party crystals-dilithium, Cryptology ePrint Archive (2022).
[77] Y. Fu, X. Zhao, Secure two-party dilithium signing protocol, in: 2021 17th International conference on computational intelligence and security (CIS), IEEE, 2021, pp. 444–448.
[78] I. Damgård, C. Orlandi, A. Takahashi, M. Tibouchi, Two-round n-out-of-n and multi-signatures and trapdoor commitment from lattices, Journal of Cryptology 35 (2) (2022) 14.
[79] S. Wuthier, J. Kim, I. Kim, S.-Y. Chang, Base station certificate and multi-factor authentication for cellular radio control communication security, arXiv preprint arXiv:2504.02133 (2025).
[80] R. Alnashwan, Y. Yang, Y. Dong, P. Gope, Strong privacy-preserving universally composable aka protocol with seamless handover support for mobile virtual network operator, in: Proceedings of the 2024 on ACM SIGSAC, 2024, pp. 2057–2071.
[81] Y. Wang, Z. Zhang, Y. Xie, {Privacy-Preserving} and {Standard-Compatible}{AKA} protocol for 5g, in: 30th USENIX security symposium (USENIX security 21), 2021, pp. 3595–3612.
[82] Z. G. Al-Mekhlafi, M. A. Al-Shareeda, B. A. Mohammed, A. A. Alsadhan, Post-quantum lattice-based forward-secure authentication scheme using fog computing in 5g-assisted vehicular networks (2024).
[83] L. K. Grover, A fast quantum mechanical algorithm for database search, in: Proceedings of the 28th ACM symp. on The. of comp., 1996.