Authentication Against Insecure Bootstrapping for 5G Networks: Feasibility, Resiliency, and Transitional Solutions in Post-Quantum Era

Saleh Darzi*, Mirza Masfiqur Rahman†, Imtiaz Karim‡, Rouzbeh Behnia§, Attila A Yavuz*, and Elisa Bertino†

***Abstract*—The 5G protocol lacks a robust base station authentication mechanism during the initial bootstrapping phase, leaving it susceptible to threats such as fake base station attacks. Conventional solutions, including digital signatures based on Public Key Infrastructures (PKIs) and identity-based signatures, are inadequate against quantum-capable adversaries. While integrating NIST's Post-Quantum Cryptography (PQC) standards is a leading approach for quantum resistance, their suitability for 5G base station authentication remains unexplored. Moreover, current solutions are predominantly centralized and lack security features such as distributed authentication.**

This work presents, to our knowledge, the first comprehensive network-level performance characterization of integrating NIST-PQC standards and conventional digital signatures (including threshold and identity-based schemes) into 5G base station authentication. Our findings reveal significant feasibility concerns, with direct PQC adoption hindered by protocol constraints and large signature sizes. We also highlight the performance limitations of conventional methods due to the overhead of certificate chains. To mitigate these challenges, we propose BORG, a transitional authentication solution based on a Hierarchical Identity-Based Threshold Signature scheme with a Fail-Stop property. BORG offers post-mortem post-quantum forgery detection and distributed trust via threshold and compact signatures, well-suited for 5G's stringent requirements. Our performance analysis underscores an important warning on the infeasibility of direct PQC integration and positions BORG as an effective transitional solution toward future quantum-resilient 5G authentication.

***Index Terms*—5G Cellular Networks, Authentication, Network Performance Analysis, Transitional Post-Quantum Security.**

I. 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 5G networks. The bootstrapping protocol enables UEs to connect to BSs and access the core network. However, initial connections at the Radio Resource Control (RRC) layer rely solely on signal strength and service parameters [1], without any authentication. This absence of BS authentication leads to severe security attacks, such as fake base stations, phishing, and spoofed emergency alerts [2], [3].

A. Prior Works on BS Authentication

Recent efforts have introduced various solutions, including Public Key Infrastructure (PKI) mechanisms [2], [4]–[6], token-based authentication methods [7], and broadcast authentication using Identity-Based Signatures (IBSs) [8]. 3GPP, the de facto standard for mobile communications, has explored PKI-based solutions for UE-to-BS authentication [9]. Among these solutions, Ross et al. [6] proposed a broadcast verification mechanism for authenticating System Information Block (SIB) messages, particularly SIB_1, which is critical in the initial RRC exchange (BS-to-UE). Hussain et al. [2] attached signatures and certificate chains to SIB_1 and SIB_2, while Lotto et al. [7] utilized an authentication token to verify the authenticity of BSs. Singla et al. [8] introduced an efficient broadcast authentication relying on a hierarchical IBS scheme. Building on these solutions, various optimizations have been proposed, including efficient certificate delivery [10], online-offline signatures [11], and outsourced computation via auxiliary entities [12]. These methods aim to reduce the signing and certificate overheads [2]. Recent efforts have focused on certificate-free methods based on hierarchical IBSs [13], [14] to improve efficiency. Below, we outline critical research gaps and the security requirements of a 5G BS authentication scheme.

(i) Feasibility of Adopting Post-Quantum Standard Measures: Most, if not all, previous studies on BS authentication rely on classical cryptographic methods (e.g., digital signatures) to ensure secure and authenticated communication [15]. However, emerging quantum computers can render classical cryptographic schemes insecure, thus posing a significant challenge to current security measures. In response, global initiatives, led by the National Institute of Standards and Technology (NIST), have focused on standardizing Post-Quantum Cryptography (PQC) [16]. The European Telecommunications Standards Institute (ETSI) and IEEE Standards Association have likewise emphasized the urgency of developing quantum-safe solutions for cellular networks and recommended practices for PQC migration [17], [18].

While preliminary integration of NIST-PQC algorithms into various network protocols such as TLS [19] and Post Quantum (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,

*Saleh Darzi and Attila A Yavuz are with the Bellini College of AI, Cybersecurity and Computing, University of South Florida; Email: salehdarzi@usf.edu; attilaayavuz@usf.edu

†Mirza Masfiqur Rahman and Elisa Bertino are with the Department of Computer Science, Purdue University, Email:rahman75@purdue.edu; bertino@purdue.edu

‡Imtiaz Karim is with the Department of Computer Science, University of Texas at Dallas, Email:imtiaz.karim@utdallas.edu

§Rouzbeh Behnia is with the School of Information Systems at University of South Florida, Email:behnia@usf.edu

due to their inherent characteristics and building blocks, the adoption of NIST-PQC schemes is far more challenging. This is further intensified by 5G's strict constraints, where initial frame synchronization messages like SIB_1 are limited to 372 bytes and are transmitted periodically with a delay as high as 160 ms (discussed in detail in Section VI). To our knowledge, there is no comprehensive evaluation or design on the adoption of PQ-secure digital signatures in 5G UE and BS authentication, which captures the protocol-level intricacies of these signatures.

(ii) Lack of Efficient Distributed and Accountable Authentication: None of the prior efforts for BS authentication account for compromised BSs, a growing concern in 5G where higher deployment densities and greater physical accessibility significantly increase the risk of compromise through various exploits [20]–[22]. Thus, an effective authentication solution must provide distributed trust. This can be achieved through threshold signatures (e.g., [23]), where multiple BSs collaboratively generate signatures, eliminating single points of failure and reducing the risk of BS or key compromises [24]. However, only a few efforts [11], [25] have explored threshold signatures in this context. Furthermore, a relevant gap lies in the lack of a distributed logging mechanism, which is also essential for detecting security breaches and conducting post-mortem analysis, particularly against quantum-capable adversaries [26]. Thereby, the authentication solution must maintain accountability by keeping verifiable logs, even in the presence of compromised BSs. Finally, resource constraints on UEs, limited packet sizes, and frequent broadcasts, especially in distributed BS authentication settings, make efficiency crucial. Therefore, the designed authentication scheme must minimize signature, communication, and storage overhead while ensuring fast signing and verification.

Scheme	E2E Delay	Comm. Cost	System Arch.	Security Features
ML-DSA [27]	5282.47	12276	2-level Cert.	Module-Lattice
Schnorr-HIBS [8]	1.61	144	Hierarchical	ECDLP
Centralized-BORG	1.64	144	Hierarchical	ECDLP/FS-PM
(t,n)-BORG	2.99	144	Hierarchical	ECDLP/FS-PM/TH

TABLE I: Comparison of signatures for 5G initial bootstrapping. Communication overhead is in bytes, and End-to-End (E2E) delay is in milliseconds. FS-PM: Fail-Stop mechanism with Post-Mortem detection. TH denotes threshold security and (t, n) is set to $(2, 3)$.

B. Our Contribution

To address these gaps, 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. To overcome these challenges, we introduce a novel transitional framework, BORG, which leverages conventional cryptography for efficiency while providing threshold (distributed) authentication, (post-mortem) forgery detection, and auditing capabilities against PQ adversaries. Our key contributions are:

(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. As SIB_1 carries essential information about BS communication and schedules future SIB messages, we identify it as the most critical message to authenticate. Our findings reveal that directly applying NIST-PQC signatures is impractical, as outlined in TABLE I and detailed in Section III. Specifically, the large signature and certificate sizes of NIST's primary (lattice-based) PQC standard *ML-DSA* [27] 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 efficient [8], these lack support for distributed authentication and PQ forgery detection.

(ii) Transitional Solution with PQ Forgery Detection and Threshold Authentication for 5G: Given the unsuitability of NIST-PQC signatures for 5G BS authentication, we propose BORG, an efficient transitional authentication framework that introduces a Hierarchical Identity-Based Threshold Signature with Fail-Stop property. *(i)* BORG provides conventional security alongside a PQ-secure fail-stop (FS) mechanism with post-mortem (PM) forgery detection, enabling computationally bounded signers to identify and prove forgeries against quantum-capable adversaries. *(ii)* BORG enables distribution of trust via threshold signatures, where the authentication system remains secure even if a subset of BSs misbehave or are compromised. *(iii)* BORG's PQ forgery detection is also reinforced by a distributed audit logging mechanism secured via PQ-secure threshold signatures, enhancing overall resiliency. *(iv)* While providing these security features, BORG maintains low communication overhead, minimal computational load on UE, and reduced end-to-end delay, eliminating the need for fragmentation. Compared to the existing conventional-secure alternatives like *Schnorr-HIBS* (see TABLE I), BORG achieves similar overhead while providing distributed authentication and PQ forgery detection.

(iii) Open-Sourced Evaluation Framework: We fully implemented BORG by incorporating it into a real 5G tested 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. Our results show that BORG is up to three orders of magnitude faster and incurs $85\times$ less communication overhead than NIST-PQC's *ML-DSA* [27]. Compared to IBS schemes [8], BORG achieves similar runtime while also providing distributed authentication with post-mortem PQ forgery detection. Its compactness, efficiency, and distributed authentication make BORG a practical transitional solution for PQ-secure 5G bootstrapping authentication. We have released the complete source code of BORG to support reproducibility.

`github.com/TheSalehDarzi/BORG-Scheme`

II. Preliminaries, Building Blocks, and Models

Notations: The symbol $||$ denotes concatenation, and $\cdot$ 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 group of prime order p with generator g. We define two cryptographically secure hash functions: $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_q$ and $H_2 : \{0,1\}^* \rightarrow$

$\mathbb{Z}_q$. $x \xleftarrow{\$} \mathcal{S}$ indicates that x is sampled uniformly at random from the set $\mathcal{S}$. 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. System Model: 5G Cellular Network

1) Network Components: The 5G cellular network consists of three main entities [28]:

- ***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.

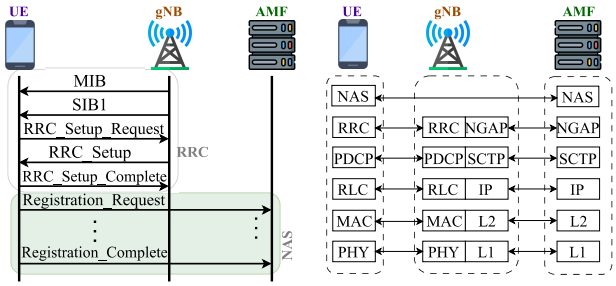

Fig. 1: Initial 5G network connection setup and protocol stack.

*2) Initial **BS-UE** Communication:* In 5G protocol stack (Fig. 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: SIB_1. Broadcast over the Downlink Shared Channel (DL-SCH), SIB_1 contains scheduling and availability information for other SIB messages. As per 3GPP specifications [29], SIB_1 has a maximum size of 372 bytes and is transmitted periodically every 160 ms, with repeated broadcasts allowed. Fig. 2 illustrates the structure of SIB_1. 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 [29]. After receiving SIB_1, the UE initiates the RRC setup. Upon successful RRC connection, the UE initiates NAS registration with the AMF.

Several additional SIB messages (SIB_2–SIB_{21}) are transmitted over DL-SCH in periodic windows, each serving specific functions. For instance, SIB_3 provides NR *intra*-frequency neighbor cell lists and reselection areas, SIB_4 conveys *inter*-frequency equivalents, SIB_9 delivers GPS and UTC time, and SIB_{15} carries disaster roaming configurations.

B. 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 four 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 [30], [31] 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 [32].
- **Key Compromise Scenarios.** We account for active adversaries capable of compromising BSs to extract signing keys [24], forge signatures, and impersonate legitimate BSs during 5G bootstrapping [20]. 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 [33].
- **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) [34].

Our threat model also captures provable detection of quantum-capable adversaries with the potential to break conventional signatures [35]. While our scheme does not provide real-time PQ security, it enables post-mortem 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 [36], [37]. This FS mechanism halts the system upon security breaks, minimizing damage and further exploitation. Since forgery detection relies

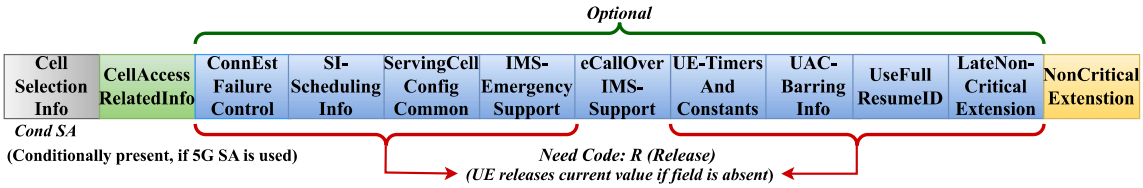

Fig. 2: SIB_1 message structure in 5G.

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

Fig. 3: Outline of Our Threat Models.

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 IV-C4).

Scope. Our objective is to design an authentication framework for 5G *UE-BS* communication. Since SIB_1 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 SIB_1 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 [38], [39]), as BORG targets the broadcast bootstrapping plane (SIB_1 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 SIB_1 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.

C. 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) [8], [40] and FROST [23]. 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 [12], [40], [41]. 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 [42]. In addition, BORG integrates an FS security mechanism [43], [44], 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 mechanism, halting further key usage and providing cryptographic evidence to absolve the signer of liability.

Definition II.1. *A hierarchical identity-based threshold signature scheme with fail-stop property is a 7-tuple algorithm as shown below:*

- $(sk_{ID_0}, PK_{ID_0}, params) \leftarrow \texttt{HITFS.Setup}(1^\kappa)$*: 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.*
- $(\{sk_{ID_{k,i}}\}_{i=1}^n, \vec{Q}_{ID_k}) \leftarrow \texttt{HITFS.Extract}(\vec{ID}_k, \vec{Q}_{ID_{(k-1)}}, sk_{ID_{(k-1)}})$*: 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 $\vec{Q}_{ID_k} = (Q_{ID_1}, Q_{ID_2}, \ldots, Q_{ID_k})$.*
- $\mathcal{L}_i \leftarrow \texttt{HITFS.Preprocess}(J)$*: 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 $\mathcal{L}_i \leftarrow (i, \{E_{i,j}\}_{j=1}^J, \{D_{i,j}\}_{j=1}^J)$.*
- $\sigma_{k,j} \leftarrow \texttt{HITFS.Sign}(m_j, \mathcal{L}_i, \{sk_{k,i}\}_{i=1}^\beta)$*: 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.*
- $\{0, 1\} \leftarrow \texttt{HITFS.MVerify}(m_j, \vec{ID}_k, \vec{Q}_{ID_k}, \sigma_{k,j})$*: 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*$(\{\hat{e}_{i,\mathrm{j}}\}_{i=1}^{\beta}, \{\hat{d}_{i,\mathrm{j}}\}_{i=1}^{\beta}, m, \sigma'_k, hist)$*: 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 II.2. *Discrete Logarithm Problem (DLP): 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 transitional 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 [45]. 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 [45].

D. Security Model

Following the hierarchical IBS [41], [46] and Schnorr-based (threshold) signature security models [8], [23], 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 forger $\mathcal{F}$ and challenger $\mathcal{C}$. The forger $\mathcal{F}$ 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 II.3. *The EUF-sID-CMIA experiment $Expt_{HITFS}^{EUF\text{-}sID\text{-}CMIA}$ for a HITFS signature scheme is defined as follows:*

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

$\mathcal{F}$ wins the experiment if the forged signature passes verification ($1 \leftarrow$ 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 $Pr[Expt_{\mathrm{HITFS}}^{EUF\text{-}sID\text{-}CMIA}(\mathcal{F}) = 1]$.

Following the principles of fail-stop signature schemes [36], [37], 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 ($EUF\text{-}sID\text{-}CMIA$, Definition II.3); 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 [43], [47], [48].

Definition II.4. *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:*

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

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

Definition II.5. *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:*

$$\Pr\begin{bmatrix} 1 \leftarrow \textit{HITFS.MVerify}(m, \vec{ID}_k, \vec{Q}_{ID_k}, \sigma_k) \wedge \\ \pi^* \leftarrow \textit{HITFS.PoF}(\{\hat{e}_{i,\mathrm{j}}^*\}_{i=1}^t, \{\hat{d}_{i,\mathrm{j}}^*\}_{i=1}^t, m, \sigma_k^*, hist) \end{bmatrix} \leq negl(\lambda_2)$$

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 $(\{\hat{e}_{i,\mathrm{j}}^*\}_{i=1}^t, \{\hat{d}_{i,\mathrm{j}}^*\}_{i=1}^t)$ of any uncorrupted signers.

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

A. Challenges of Deploying NIST-PQC Signatures

For any BS authentication mechanism, it is critical that the BS signs the SIB_1 message and, ideally, embeds the signature within the same packet to avoid additional overhead [2]. However, including the signature in SIB_1 is challenging due to strict size constraints. According to the 3GPP RRC specification [29], the maximum allowable size for the SIB_1 message is 2976 bits or 372 bytes (section 5.2.1). However, for initial communication, not all the bytes of the SIB_1 are used.

For instance, a minimally configured SIB_1 typically occupies 80–100 bytes. To assess real-world usage and configuration, we take measurements for SIB_1 messages from two major U.S. vendors across multiple BSs–covering 8 different physical cells. The observed SIB_1s 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 III-B).

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 SIB_1. For example, the *FN-DSA* [49] 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* [50] 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* [27]) produces a 2420-byte signature and a 1312-byte public key. These sizes far exceed the SIB_1 limit, and the only way to transmit such signatures would be to fragment both the signature and public key across multiple SIB_1 packets.

B. 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 SIB_1, the BS would need to send 13 separate SIB_1 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 SIB_1 packets. However, the latency implications further exacerbate this challenge. According to the RRC specification, "*The SIB_1 is transmitted on the DL-SCH with a periodicity of* **160 *ms*** *and variable transmission repetition periodicity within* **160 *ms*** *... The default transmission repetition periodicity of SIB_1 is* **20 *ms*** *but the actual transmission repetition periodicity is up to network implementation.*" (section 5.2.1 [29]). This implies a default delay of 20 ms between consecutive SIB_1 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 $20 \cdot 12 = 240$ ms upto $160 \cdot 12 = 1920$ ms, not including the time for packet generation at the BS and reconstruction at the UE. Moreover, the delivery of SIB_1 can be unreliable over 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 19, resulting in an overall transmission delay ranging from $20 \cdot 18 = 360$ ms upto $160 \cdot 18 = 2880$ 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 SIB_1s. 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 SIB_1 packets adds significant complexity to lower protocol layers. For example, in the srsRAN open-source 5G stack, even when multiple SIB_1 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 SIB_1 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.

C. 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 $B210$ and srsgNB on another USRP $B210$–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. Figure 4 shows our complete testbed setup.

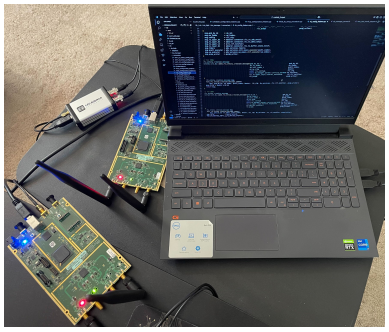

Fig. 4: Testbed setup for 5G end-to-end communication. The gNB (USRP: bottom-left) and UE (USRP: top-right) are connected to a laptop and GPSDO.

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 III), 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 VI.

IV. The Proposed Transitional Solution

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

A. 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 $i = 1, \ldots, n$ 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 [51] 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.

1: **for** $i = 1, 2, \ldots, n$ **do**
2: **for** $j = 1, 2, \ldots, J$ **do**
3: $(\hat{e}_{i,j}, \hat{d}_{i,j}) \xleftarrow{\$} \mathbb{Z}_q \times \mathbb{Z}_q$
4: $e_{i,j} \leftarrow H_1(\hat{e}_{i,j} || j || ID_{k,i})$, $d_{i,j} \leftarrow H_1(\hat{d}_{i,j} || j || ID_{k,i})$
5: $E_{i,j} \leftarrow g^{e_{i,j}} \mod p$ and $D_{i,j} \leftarrow g^{d_{i,j}} \mod p$
6: Send $(\{E_{i,j}\}_{j=1}^J, \{D_{i,j}\}_{j=1}^J)$ to the other $n-1$ users.
7: **for** $i = 1, 2, \ldots, n$ **do**
8: **if** $(\{E_{i,j}\}_{j=1}^J, \{D_{i,j}\}_{j=1}^J) \in \mathbb{G}$ **then**
9: $\mathcal{L}_{i,j} \leftarrow (E_{i,j}, D_{i,j})$
10: **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., [23]), 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 $i \in \{1, \ldots, n\}$ 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 $j \in \{1, \ldots, J\}$, 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 IV-B.

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 [43]. Leveraging this principle and following fail-stop mechanisms (e.g., [43], [44]), 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.

BORG.PoF (Algorithm 4) is invoked by signers at level k upon detecting a suspected forgery of message m_j. 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 preprocess-

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 $i = 1, 2, \ldots, \beta$ **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 $i = 1, 2, \ldots, \beta$ **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 $\ell = 1, 2, \ldots, k$
$Q \leftarrow \prod_{\ell=1}^{k-1} (Q_{ID_\ell})^{\prod_{\omega=\ell+1}^{k} h_{ID_\omega}}$
$h_j \leftarrow H_2(R_j||Q_{ID_k}||m_j)$
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

ing. 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 $\pi = (\mathrm{j}, \{\hat{e}_{i,\mathrm{j}}\}_{i=1}^{\beta}, \{\hat{d}_{i,\mathrm{j}}\}_{i=1}^{\beta})$, 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.

B. Signature Verification Correctness

We demonstrate the correctness of the verification procedure by proving that an honestly generated signature in the 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:

Algorithm 4 BORG *(Forgery Proof 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 $k-1$:

$\mathrm{j} \leftarrow hist$ 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*" **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

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

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:

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

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.

C. 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.

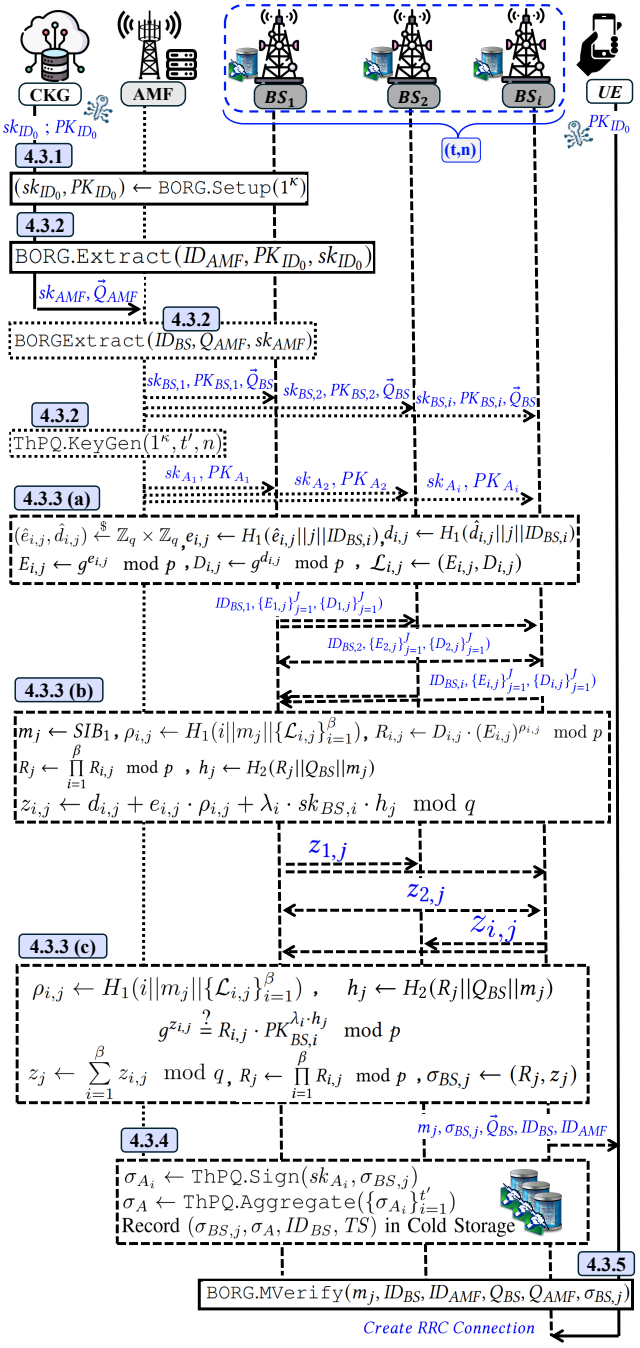

Fig. 5: Our protocol for authenticating 5G cellular BSs.

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 [52]. 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 [26], [53], [54]. 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 post-mortem forgery detection.

The UE verifies the SIB_1 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 fail-stop 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.

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.

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.

3) Signing Broadcast Messages Phase: To sign SIB_1, $\beta \in [t, n]$ 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 $(m_j, \sigma_{BS,j}, \vec{Q}_{BS}, ID_{BS}, ID_{AMF})$.

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 SIB_1 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 faul-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) Signature Verification Phase: The signature verification process follows `BORG.MVerify` as specified in Algorithm 3. Given a signature on the SIB_1 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.

6) Forgery Proof and Verification Phase: Each BS has access to the authenticated distributed cold storage that records all SIB messages and aggregated signatures. Upon detecting or suspecting forgery, a BS initiates the proof-of-forgery protocol with the other t participating signers, as illustrated in Fig. 6. For a suspected forged signature σ'_{BS} on a SIB_1 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 the AMF and the core network to halt the system.

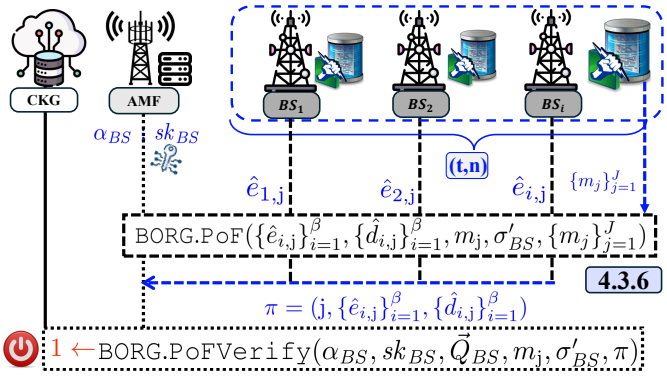

Fig. 6: Instantiation of Forgery Proof and Verification Phase.

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.

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, 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 as long as 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 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 [55], 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.

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.

V. Security Analysis

Following Definition II.3, we prove *EUF-sID-CMIA* security of BORG under the hardness of the (Elliptic Curve)-DLP based on the generalized forking lemma in the random oracle model [56], [57].

Theorem 1. *If a forger $\mathcal{F}$ can $(q_E, q_P, q_S, q_{H_1,H_2})$-break BORG in the random oracle model (Definition II.3) 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 $\mathbb{G}$.*

Proof. We assume that the forger $\mathcal{F}$ 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. $\mathcal{F}$ has control over $(t-1)$ signing participants. We consider a challenger $\mathcal{C}$ which invokes $\mathcal{F}$ 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 $\vec{ID}^*$ chosen by the forger. $\mathcal{F}$ starts by picking a target identity $\vec{ID}^* = (ID_1^*, \ldots, ID_\ell^*) \in {\mathbb{Z}_p}^\ell$ 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 $\mathcal{F}$ while keeping (α_0, sk_{ID_0}) secret.

• **Execute $\mathcal{F}^{\mathcal{O}_E,\mathcal{O}_P,\mathcal{O}_S}(PK_{ID_0}, params)$:** $\mathcal{F}$ 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 $\vec{ID} = (ID_1, \ldots, ID_\ell) \in {\mathbb{Z}_p}^\ell$, where $\vec{ID} \neq \vec{ID}^*$ or any $ID_i^* \notin \vec{ID}$ for $i = 1, \ldots, \ell$, $\mathcal{C}$ 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 $i = 1, \ldots, t-1$. 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 $i = 1, \ldots, \ell$), 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 $i = 1, \ldots, t-1$, following BORG.Preprocess(.).
- *Signing Oracle ($\mathcal{O}_S$):* For a message m and $\vec{ID} = (ID_1, \ldots, ID_\ell) \in {\mathbb{Z}_p}^\ell$, where $\vec{ID} \neq \vec{ID}^*$ or any $ID_i^* \notin \vec{ID}$ for $i = 1, \ldots, \ell$, 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 $\mathcal{F}$:*** $\mathcal{F}$ produces a valid signature (m^*, σ^*) for ID^* under the master and group public keys (PK_{ID_0}, Q_{ID_k}). $\mathcal{F}$ wins the experiment if satisfies the conditions mentioned in Definition II.3.

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

- Applying GFL, the forger $\mathcal{F}$ 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 $\mathbb{G}$. □

Following Definition II.4, we prove the signer-side fail-stop security of BORG under the hardness of second preimage resistance of a cryptographically secure hash function.

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 II.4, and λ_2-bit non-repudiation security against quantum adversaries with access to one-out-oft signing participants, as captured in Definition II.5. 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 II.3.*

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 II.4) with non-negligible advantage ϵ, producing a forged signature σ_k^* on a message m^* that (i) passes verification ($1 \leftarrow \texttt{BORG.MVerify}(m^*, \vec{ID}_k, \vec{Q}_{ID_k}, \sigma_k^*)$), and (ii) cannot be proven invalid via $0 \leftarrow \texttt{BORG.PoF}(\{\hat{e}_{i,\mathrm{j}}^*\}_{i=1}^t, \{\hat{d}_{i,\mathrm{j}}^*\}_{i=1}^t, m^*, \sigma_k^*, hist)$, 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 $(\hat{e}_{i,t}^{*\prime}, \hat{d}_{i,t}^{*\prime})$ that satisfy the conditions mentioned in Definition II.4 such that $H_1(\hat{e}_{i,t}^{*\prime} || j || ID_{k,t}) = H_1(\hat{e}_{i,t}^* || j || ID_{k,t})$ and $H_1(\hat{d}_{i,t}^{*\prime} || j || ID_{k,t}) = H_1(\hat{d}_{i,t}^* || j || ID_{k,t})$ while $(\hat{e}_{i,t}^{*\prime}, \hat{d}_{i,t}^{*\prime}) \neq (\hat{e}_{i,t}^*, \hat{d}_{i,t}^{*\prime})$, then $\mathcal{C}_1$ outputs $(\hat{e}_{i,t}^{*\prime}, \hat{d}_{i,t}^{*\prime})$ 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 [58], 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 II.5, 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 II.5 with non-negligible probability ϵ; that is, $\mathcal{A}$ participates in generating a valid signature σ_k such that (i) it passes verification $1 \leftarrow \texttt{BORG.MVerify}(m, \vec{ID}_k, \vec{Q}_{ID_k}, \sigma_k)$, and (ii) later constructs a forged proof π^* satisfying $1 \leftarrow \texttt{BORG.PoFVerify}(\alpha_k, sk_{ID_{k-1}}, \vec{Q}_{ID_k}, m, \sigma_k', \pi^*)$, 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 $\hat{e}_{i,j}^{*\prime} \neq \hat{e}_{i,j}^*$ such that $H_1(\hat{e}_{i,j}^{*\prime} || j || ID_{k,i}) = H_1(\hat{e}_{i,j}^* || j || ID_{k,i})$, 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 $(\hat{e}_{i,j}^{*\prime}, \hat{d}_{i,j}^{*\prime})$ that satisfies the conditions mentioned in Definition II.5, 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 [58], the adversary's success probability reduces to $\mathcal{O}(2^{n/2})$, yielding λ_2-bit PQ security for an n-bit hash function. □

VI. Performance Evaluation

This section presents a comprehensive performance evaluation and comparison of NIST-PQC signatures, selected conventional-secure alternatives, and the proposed BORG scheme for 5G initial bootstrapping authentication.

A. 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 III. Real network packets were investigated using the Network Signal Guru Android app installed on a OnePlus Nord 5G smartphone [59].

Libraries: We employed OpenSSL library[1] for cryptographic primitives such as hash functions and EC operations (e.g., point multiplication, modular arithmetic), the Open Quantum-Safe library[2] for NIST-PQC schemes, the Ringtail library[3] for the ThPQ, and the blst[4] 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 [60]. 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 SIB_1 message, the srsRAN gNB utilizes 79 bytes out of the allowed

[1]OpenSSL Library: https://openssl-library.org/
[2]Open Quantum-Safe Library: https://openquantumsafe.org/
[3]Ringtail Library: https://github.com/daryakaviani/ringtail
[4]BLST Library: https://github.com/Chia-Network/bls-signatures

Scheme	Sign (ms)	Verif. (ms)	Packet Proc.(ms)	Transmission (ms)	Crypto./Comm.(B)	PK (B)	E2E Delay (ms)
BLS [61]	0.42	1.15	0.03	< 0.01	$48/-$	96	1.60
ML-DSA [27]	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

TABLE II: Quantitative comparison of the candidate signature schemes in an overly ideal scenario (*sending only signature*) for authenticating 5G cellular BSs. 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 SIB_1 packet. $(2,3)$-BORG* considers a precomputed preprocessing phase. †*EC-Schnorr* [62] shares identical timing and size metrics with *Schnorr-HIBS* [8].

372 bytes. Accordingly, all subsequent evaluations report the computational and communication overhead for signing a 79-byte SIB_1 message, as reflected in the tables. Note that even considering slightly larger SIB_1s observed from real networks, our results remain consistent.

B. Evaluation Metrics

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 delay. The 5G processing latency accounts for the duration of the operations for all the network entities (such as gNB or UE) to process the cryptographic material in/from the network packets without doing actual cryptographic operations. This also includes the cost of processing signature fragments whenever applicable. The qualitative evaluation considers features such as system architecture, breach resiliency, and PQ assurances.

Selection Criteria for Counterparts: For PQ counterparts, we consider NIST-PQC signatures like lattice-based *ML-DSA* [27], *FN-DSA* [49], and hash-based *SLH-DSA* [50]. Given the very large signature and execution times of hash-based alternatives (e.g., *SLH-DSA* with a 7856-byte signature, nearly $3\times$ that of *ML-DSA*), we mainly focus on *ML-DSA* for direct comparison. While *FN-DSA-1024* [49] 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., [63]), and exhibits comparable timings and sizes to *ECDSA* [64], we adopt *EC-Schnorr* [62] as the foundation for our scheme and the primary baseline for comparison. For conventionally secure signatures, we consider *EC-Schnorr* [64] as an optimized standard, *BLS* [61] 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 [6] 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* [7], 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. A detailed quantitative and qualitative comparison follows.

C. Experimental Results

TABLE II presents a quantitative comparison of candidate schemes for signing a single SIB_1 message, evaluating signing/verification time, 5G processing, cryptographic/communication overhead, and end-to-end (E2E) latency. TABLE III extends this with both qualitative and quantitative analysis in the full 5G hierarchical bootstrapping context.

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 SIB_1 message, with the results summarized in TABLE II. *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 SIB_1 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 SIB_1 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. Additionally, our protocol adopts Ringtail [45] as the ThPQ instantiation for distributed audit logging due to its performance benefits. Since ThPQ only signs the SIB_1 signatures and does not affect BS or UE operations, it is excluded from

Scheme	System Architecture and Features	Sign Delay (ms)	Full Verification Delay (ms)	5G Delay (ms)	Crypto./Comm. Overhead (B)	E2E Delay (ms)
BLS [61]	2-Level Certificate with Aggregation	0.42	3.46	0.05	$240/-$	3.93
EC-Schnorr [62]	2-level Certificate	0.30	3.80	0.05	$256/-$	4.15
ML-DSA [27]	2-Level Certificate	0.12	0.12	$662.23 \sim 5282.23$	$9884/12276$	$662.47 \sim 5282.47$
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

TABLE III: Comparison of candidate signature schemes for authenticating $SIB1$. 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 SIB_1 packet.

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 III), which accounts for transmission of keys, certificates, and IDs, BS signing costs remain consistent with those reported in TABLE II. For full verification, however, the UE must validate the SIB_1 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 SIB_1 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 SIB_1 packet, demonstrating the superior efficiency of our transitional scheme over existing NIST-PQ solutions.

• *Communication Overhead:* In 5G BS authentication, certificate-based schemes must transmit the SIB_1 message, signature, public keys, and two certificates (for the AMF and BS) to establish key authenticity. In contrast, hierarchical schemes transmit only the SIB_1 signature, corresponding public keys, and identities, eliminating certificates and reducing communication overhead. As shown in TABLE III, 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 SIB_1 packet. While the threshold-BORG requires inter-gNB communication for signature aggregation via the XnAP interface [65] (using SCTP over IP [66]), this delay is implementation-dependent and typically below 10 ms [67]. Even under this upper bound, *(2,3)-BORG* remains significantly more efficient than NIST-PQ alternatives requiring fragmentation.

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 SIB_1 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 III, even efficient, conventionally secure schemes such as *EC-Schnorr* and *BLS* incur notable cryptographic overhead. For instance, if SIB_1 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 complete 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* [23]), which resemble *Schnorr-HIBS* and $(2,3)$-BORG, incur higher computational costs when applied to SIB_1 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 [68]), homomorphic hashing and commitments (e.g., *Dilizium* [69] incurs hundreds of milliseconds and 21120-byte signatures), and fully homomorphic encryption [70], which leads to delays in the order of seconds. Additional constructions, such as those based on the Fiat–Shamir with Aborts paradigm [71] 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 III, NIST-PQC signatures are currently unsuitable for 5G SIB_1 authentication due to their large sizes, high computational costs, and substantial communication overhead. These signatures exceed the SIB packet size limit. Experimental results (TABLEs II-III) 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 SIB_1 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 quantum resilience and offer no protection against compromised BSs. BORG achieves three orders of magnitude faster execution and $85\times$ lower communication overhead than full-PQC alternatives like *ML-DSA*, while offering a fail-stop mechanism, PQ forgery detection, and breach resiliency. Its efficiency, comparable to conventional hierarchical schemes, makes it a practical solution for 5G BS authentication.

VII. Related Work

To secure 5G UE-to-BS connection bootstrapping and provide BS authentication, particularly the SIB messages, various approaches have been proposed. Early efforts build upon traditional PKI-based mechanisms discussed in 3GPP specifications [9], relying on certificate-based authentication frameworks [4], [5]. *TESLA* [72] introduced a lightweight broadcast authentication scheme based on loose time synchronization. Hussain et al. [2] attached digital signatures and certificate chains to SIB_1/SIB_2 messages, while Ross et al. [6] proposed a "broadcast but verify" model, transmitting the signature in a separate *signingSIB* message to reduce overhead. Wuthier et al. [73] proposed a multi-factor authentication with an offline blockchain-based certificate delivery system. *BARON* [7] employed symmetric tokens for pre-authentication defense and introduced the concept of Closed Trusted Entity (CTE) for secure connection initialization and handover in 5G networks.

To address centralized trust limitations, Sengupta et al. [11] explored threshold signatures in online-offline settings. Alnashwan et al. [74] presented a UC-secure authentication and handover protocol with strong user privacy, while Wang et al. [75] proposed encryption and KEM-based countermeasures for 5G-AKA linkability. Certificate-free schemes leveraging identity-based signatures have also been explored to reduce certificate overhead and enhance efficiency [8], [13], [14]. In terms of PQ security, recent work has focused on PQ-AKA protocols [38], [39], [76], [77] using lattice-based KEMs and fully homomorphic encryption [78]. Hybrid solutions combining NIST-PQC KEMs with symmetric primitives [79]–[81] aim to reduce computational overhead. However, none of these efforts explicitly address the unique constraints of initial 5G base station authentication and bootstrapping security, such as stringent message size limits and low-latency requirements.

VIII. Limitations, Conclusion, and Future Work

In conclusion, our work 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. Through a detailed feasibility assessment and network-level performance analysis, we demonstrate that existing approaches fall short of meeting 5G's efficiency, scalability, and PQ resilience requirements. To bridge this critical gap, we present BORG, a lightweight, distributed, and compromise-resilient transitional solution that offers fail-stop PQ forgery detection and compact authentication tailored for real-world 5G constraints. Our findings underscore the urgency of rethinking PQ integration in mobile networks and position BORG as a practical and transitional step toward quantum-secure 5G authentication. Note that our solution does not offer full PQ security and does not address privacy concerns related to UE-to-BS connections, passive eavesdropping, or other physical-layer threats and remains vulnerable to overshadow attacks. As it currently secures only the SIB packets, future work will focus on extending protection to mitigate overshadow attacks in the PQ era.

Acknowledgment

This work is supported by the NSF-SNSF (2444615), 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.

References

[1] S. Wuthier, J. Kim, J. Kim, and 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, and 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, and L. Xiong, "A survey on security aspects for 3gpp 5g networks," *IEEE communications surveys & tutorials*, vol. 22, no. 1, pp. 170–195, 2019.

[4] C.-C. Lee, I.-E. Liao, and M.-S. Hwang, "An extended certificate-based authentication and security protocol for mobile networks," *Information Technology and Control*, vol. 38, no. 1, 2009.

[5] 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.

[6] A. J. Ross, B. Reaves, Y. Nasser, G. Cukierman, and 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.

[7] A. Lotto, V. Singh, B. Ramasubramanian, A. Brighente, M. Conti, and 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.

[8] A. Singla, R. Behnia, S. R. Hussain, A. Yavuz, and 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, pp. 501–515.

[9] *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.

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

[11] B. Sengupta and 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.

[12] M. Ramadan, Y. Liao, F. Li, and S. Zhou, "Identity-based signature with server-aided verification scheme for 5g mobile systems," *IEEE Access*, vol. 8, pp. 51 810–51 820, 2020.

[13] Y. Dong, R. Behnia, A. A. Yavuz, and S. R. Hussain, "Securing 5g bootstrapping: A two-layer ibs authentication protocol," *arXiv preprint arXiv:2502.04915*, 2025.

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

[15] A. A. Yavuz, K. Sedghighadikolaei, S. Darzi, and S. E. Nouma, "Beyond basic trust: Envisioning the future of nextgen networked systems and digital signatures," in *2023 5th IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. IEEE, 2023, pp. 267–276.

[16] S. Darzi, K. Ahmadi, S. Aghapour, A. A. Yavuz, and 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.

[17] *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.

[18] *IEEE Standards & Projects for Quantum Technologies*, https://standards.ieee.org/initiatives/quantum-standards-activities/.

[19] D. Sikeridis, P. Kampanakis, and 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. [Online]. Available: https://www.ndss-symposium.org/ndss-paper/post-quantum-authentication-in-tls-1-3-a-performance-study/

[20] N. Bennett, W. Zhu, B. Simon, R. Kennedy, and 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*, ser. CCS '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 2027–2041.

[21] *Telecom Security Incidents 2020 - Annual Report. Technical Report. European Union Agency for Cybersecurity*, https://www.enisa.europa.eu/publications/telecom-annual-incident-reporting-2020.

[22] *The Rising Threat Landscape for Cell Towers*, https://bioconnect.com/2023/05/31/the-rising-threat-landscape-for-cell-towers/.

[23] C. Komlo and 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.

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

[25] O. Vikhrova, S. Pizzi, A. Terzani, L. Araujo, A. Orsino, and G. Araniti, "Multi-sim support in 5g evolution: Challenges and opportunities," *IEEE Communications Standards Magazine*, vol. 6, no. 2, pp. 64–70, 2022.

[26] A. A. Yavuz, "System and method for secure review of audit logs," Jun. 11 2019, uS Patent 10,318,754.

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

[28] H. Fourati, R. Maaloul, L. Chaari, and M. Jmaiel, "Comprehensive survey on self-organizing cellular network approaches applied to 5g networks," *Computer Networks*, vol. 199, p. 108435, 2021.

[29] *3GPP RRC Specification*, 2024, https://www.etsi.org/deliver/etsi_ts/138300_138399/138331/18.01.00_60/ts_138331v180100p.pdf.

[30] H. Kim, J. Lee, E. Lee, and 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.

[31] G. Lee, J. Lee, J. Lee, Y. Im, M. Hollingsworth, E. Wustrow, D. Grunwald, and S. Ha, "This is your president speaking: Spoofing alerts in 4g lte networks," in *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '19. Association for Computing Machinery, 2019, p. 404–416.

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

[33] "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, 2025, accessed: October, 2025.

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

[35] C. J. Mitchell, "The impact of quantum computing on real-world security: A 5g case study," *Computers & Security*, vol. 93, p. 101825, 2020.

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

[37] C. Boschini, H. Dahari, M. Naor, and 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.

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

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

[40] S. S. Chow, L. C. Hui, S. M. Yiu, and 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.

[41] D. Galindo and 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 21-25, 2009. Proceedings 2*. Springer, 2009, pp. 135–148.

[42] S. Ergezer, H. Kinkelin, and F. Rezabek, "A survey on threshold signature schemes," *Network*, vol. 49, 2020.

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

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

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

[46] C.-P. Schnorr, "Efficient identification and signatures for smart cards," in *Advances in Cryptology—CRYPTO'89 Proceedings 9*. Springer, 1990, pp. 239–252.

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

[48] J. J.-R. Chen, Y.-Y. Chiang, W.-H. Hsu, and W.-Y. Lin, "Fail-stop group signature scheme," *Security and Communication Networks*, no. 1, 2021.

[49] D. Soni, K. Basu, M. Nabeel, N. Aaraj, M. Manzano, and R. Karri, "Falcon," *Hardware Architectures for Post-Quantum Digital Signature Schemes*, pp. 31–41, 2021.

[50] D. Cooper *et al.*, "Stateless hash-based digital signature standard," 2024.

[51] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[52] *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.

[53] S. E. Nouma and 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.

[54] T. Le, P. Huang, A. A. Yavuz, E. Shi, and T. Hoang, "Efficient dynamic proof of retrievability for cold storage," *Cryptology ePrint Archive*, 2022.

[55] C. Wendt and M. Barnes, "Rfc 8588: Personal assertion token (passport) extension for signature-based handling of asserted information using tokens (shaken)," 2019.

[56] M. Bellare and 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.

[57] A. Boldyreva, A. Palacio, and B. Warinschi, "Secure proxy signature schemes for delegation of signing rights," *Journal of Cryptology*, vol. 25, pp. 57–115, 2012.

[58] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the 28th ACM symp. on Theory of comp.*, 1996.

[59] *Network Signal Guru User Manual*, https://m.qtrun.com/docs/NSG_Manual_Aug_2017.pdf.

[60] G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, and Y.-K. Liu, "Status report on the third round of the nist post-quantum cryptography standardization process," 2022.

[61] D. Boneh, B. Lynn, and 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.

[62] C.-P. Schnorr, "Efficient signature generation by smart cards," *Journal of cryptology*, vol. 4, pp. 161–174, 1991.

[63] G. Maxwell, A. Poelstra, Y. Seurin, and P. Wuille, "Simple schnorr multi-signatures with applications to bitcoin," *Designs, Codes and Cryptography*, vol. 87, no. 9, pp. 2139–2164, 2019.

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

[65] *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.
[66] *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.
[67] R. Stewart, "Stream control transmission protocol," Tech. Rep., 2007.
[68] D. Cozzo and N. P. Smart, "Sharing the luov: threshold post-quantum signatures," in *IMA International Conference on Cryptography and Coding*. Springer, 2019, pp. 128–153.
[69] P. Laud, N. Snetkov, and J. Vakarjuk, "Dilizium 2.0: Revisiting two-party crystals-dilithium," *Cryptology ePrint Archive*, 2022.
[70] Y. Fu and X. Zhao, "Secure two-party dilithium signing protocol," in *2021 17th International conference on computational intelligence and security (CIS)*. IEEE, 2021, pp. 444–448.
[71] I. Damgård, C. Orlandi, A. Takahashi, and M. Tibouchi, "Two-round n-out-of-n and multi-signatures and trapdoor commitment from lattices," *Journal of Cryptology*, vol. 35, no. 2, p. 14, 2022.
[72] A. Perrig, J. D. Tygar, A. Perrig, and J. Tygar, "Tesla broadcast authentication," *Secure Broadcast Communication: In Wired and Wireless Networks*, pp. 29–53, 2003.
[73] S. Wuthier, J. Kim, I. Kim, and S.-Y. Chang, "Base station certificate and multi-factor authentication for cellular radio control communication security," *arXiv preprint arXiv:2504.02133*, 2025.
[74] R. Alnashwan, Y. Yang, Y. Dong, and 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.
[75] Y. Wang, Z. Zhang, and Y. Xie, "{Privacy-Preserving} and {Standard-Compatible}{AKA} protocol for 5g," in *30th USENIX security symposium (USENIX security 21)*, 2021, pp. 3595–3612.
[76] M. T. Damir and V. Niemi, "On post-quantum identification in 5g," in *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2022, pp. 292–294.
[77] Z. G. Al-Mekhlafi, M. A. Al-Shareeda, B. A. Mohammed, and A. A. Alsadhan, "Post-quantum lattice-based forward-secure authentication scheme using fog computing in 5g-assisted vehicular networks," 2024.
[78] J. Cao, P. Yu, X. Xiang, M. Ma, and H. Li, "Anti-quantum fast authentication and data transmission scheme for massive devices in 5g nb-iot system," *IEEE Internet of Things Journal*, pp. 9794–9805, 2019.
[79] R. C. Vuppala, D. Kumar, D. Je, N. Sharma, A. Nigam, and 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.
[80] Y. Ko, I. Pawana, and 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.
[81] P. Scalise, R. Garcia, M. Boeding, M. Hempel, and H. Sharif, "An applied analysis of securing 5g/6g core networks with post-quantum key encapsulation methods," *Electronics*, vol. 13, no. 21, p. 4258, 2024.

Saleh Darzi is a Ph.D. Candidate in the Bellini College of Artificial Intelligence, Cybersecurity, and Computing, actively engaged in research within the Applied Cryptography Research Laboratory (ACRL) under the supervision of Dr. Attila Yavuz at the University of South Florida. His primary research pursuits revolve around post-quantum and applied cryptography, with a focus on addressing challenges in the privacy and security of IoT, Blockchain technology, and network security. Saleh holds a Master of Science degree in Electrical Engineering (Communication-System) from K. N. Toosi University of Technology, Tehran, Iran, obtained in 2021. He is a member of IEEE and ACM.

Mirza Masfiqur Rahman is a Ph.D. student in the Cyber Space Security Lab (cyber2SLab) at Purdue University, working with Dr. Elisa Bertino. His primary research is on the security and privacy of network systems, including 4G/5G, and Open-RAN. He combines protocol verification, natural language processing, formal methods, and property-based testing to reduce exploitable ambiguities, strengthen trust in UE-RAN-Core interactions, and inform secure deployment practices for operators and vendors. He holds a B.Sc. from the Bangladesh University of Engineering and Technology (BUET). He is a member of ACM.

Imtiaz Karim is an Assistant Professor in the Department of Computer Science at the University of Texas at Dallas. Before that, he was a Postdoctoral Researcher in the Department of Computer Science at Purdue University. He completed his Ph.D. from the same department in Spring 2023. He leads the System and Network Security (SysNetS) lab at UTD. Dr. Karim's research lies in the general area of systems and network security. More specifically, the focus is on ensuring the security and privacy of wireless communication protocols (e.g., cellular networks-4G/5G, Bluetooth, VoWiFi, vehicular, WiFi, and IoT) with respect to their design and implementation. His research has led to several changes in the design of 4G and 5G cellular standards. He has received acknowledgments from GSMA Mobile Security Research, the WiFi Alliance, Google, Qualcomm, Samsung, MediaTek, Huawei, and other vendors. He received the Best Paper award at ACSAC 2019 and the Best Paper award nomination at ICDCS 2021. He is a member of IEEE and ACM.

Rouzbeh Behnia is an assistant professor at the School of Information Systems at the University of South Florida. His research focuses on different aspects of cybersecurity and applied cryptography. He is particularly interested in addressing privacy challenges in AI systems, developing post-quantum cryptographic solutions, and enhancing authentication protocols to ensure the integrity of computation and communication.

Attila A Yavuz is an Associate Professor at the Bellini College of Artificial Intelligence, Cybersecurity, and Computing at the University of South Florida (USF), where he also directs the Applied Cryptography Research Laboratory. Previously, he was an Assistant Professor at Oregon State University (2014–2018) and USF (2018–2021), following his role as a research scientist at the Robert Bosch Research and Technology Center North America (2011–2014). He holds a Ph.D. in Computer Science from North Carolina State University (2011) and an M.S. from Bogazici University (2006). Dr. Yavuz's broad research interests center on designing, analyzing, and deploying cryptographic techniques to strengthen the security of computer systems and next-generation networks. His work has been recognized with numerous honors, including the NSF CAREER Award, multiple research awards from Bosch (five) and Cisco (four), three USF Excellence in Research Awards, several major federal grants, and numerous best paper awards. His research leadership extends to editorial board service (e.g., IEEE TDSC) and organizing roles in major conferences (e.g., ACM CCS). His work encompasses 115 peer-reviewed publications in top-tier venues (e.g., Usenix, NDSS, CCS, IEEE TIFS), patents, and technology transfers to industry partners, particularly in searchable encryption and intra-vehicular network security, impacting tens of millions of users worldwide. He is a Senior Member of the IEEE, the National Academy of Inventors, and ACM.

Elisa Bertino is a Samuel D. Conte Distinguished Professor of Computer Science at Purdue University. Prior to joining Purdue in 2004, she was a Professor and department head at the Department of Computer Science and Communication of the University of Milan. She has been a postdoc at the IBM Research Laboratory (now Almaden) in San Jose, and a Visiting Professor at the Singapore National University and the Singapore Management University. At Purdue she served as Research Director of the Center for Education and Research for Information Assurance and Security (CERIAS) (2004-2015) and Director of Discovery Park CyberCenter (2010-2016). She is a Life Fellow member of IEEE, ACM, and AAAS. She received the 2002 IEEE Computer Society Technical Achievement Award, the 2005 IEEE Computer Society Tsutomu Kanai Award for "Pioneering and innovative research contributions to secure distributed systems", the 2019-2020 ACM Athena Lecturer Award, the 2021 IEEE Innovation in Societal Infrastructure Award, and the 2025 ACM SIGSAC Outstanding Innovation Award. She has worked for more than 40 years in data security and privacy. Recently she has been working on security of cellular networks, mobile applications and IoT systems, zero-trust architectures, and machine learning techniques for cybersecurity. She served as EiC of the IEEE Transactions on Dependable and Secure Computing and as chair of ACM SIGSAC. She is currently serving as ACM Vice-President.