Look Before You Leap: Secure Connection Bootstrapping for 5G Networks to Defend Against Fake Base-Stations

Ankush Singla
Department of Computer Science
Purdue University
West Lafayette, Indiana, USA
asingla@purdue.edu

Rouzbeh Behnia
Department of Computer Science and Engineering
University of South Florida
Tampa, Florida, USA
behnia@usf.edu

Syed Rafiul Hussain
Department of Computer Science and Engineering
The Pennsylvania State University
Pennsylvania, USA
hussain1@psu.edu

Attila A. Yavuz
Department of Computer Science and Engineering
University of South Florida
Tampa, Florida, USA
attilaayavuz@usf.edu

Elisa Bertino
Department of Computer Science
Purdue University
West Lafayette, Indiana, USA
bertino@purdue.edu

ABSTRACT

The lack of authentication protection for bootstrapping messages broadcast by base-stations makes impossible for devices to differentiate between a legitimate and a fake base-station. This vulnerability has been widely acknowledged, but not yet fixed and thus enables law-enforcement agencies, motivated adversaries and nation-states to carry out attacks against targeted users. Although 5G cellular protocols have been enhanced to prevent some of these attacks, the root vulnerability for fake base-stations still exists. In this paper, we propose an efficient broadcast authentication protocol based on a hierarchical identity-based signature scheme, Schnorr-HIBS, which addresses the root cause of the fake base-station problem with minimal computation and communication overhead. We implement and evaluate our proposed protocol using off-the-shelf software-defined radios and open-source libraries. We also provide a comprehensive quantitative and qualitative comparison between our scheme and other candidate solutions for 5G base-station authentication proposed by 3GPP. Our proposed protocol achieves at least a 6x speedup in terms of end-to-end cryptographic delay and a communication cost reduction of 31% over other 3GPP proposals.

CCS CONCEPTS

• **Security and privacy** → **Security protocols**; **Digital signatures**; **Mobile and wireless security**.

KEYWORDS

5G; Cellular Networks; Fake Base-Stations

ACM Reference Format:
Ankush Singla, Rouzbeh Behnia, Syed Rafiul Hussain, Attila A. Yavuz, and Elisa Bertino. 2021. Look Before You Leap: Secure Connection Bootstrapping for 5G Networks to Defend Against Fake Base-Stations. In *2021 ACM Asia Conference on Computer and Communications Security (ASIA CCS '21), June 7–11, 2021, Hong Kong, Hong Kong*. ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3433210.3453082

1 INTRODUCTION

Fifth-generation (5G) cellular networks provide faster connectivity, greater bandwidth, and better security postures than previous generations of cellular technologies. While many such improvements can be largely attributed to underlying physical layer communication technologies and new security policies in the upper layer communication, 5G inherits many mechanisms from previous generations because of backward compatibility. One such mechanism is the cell (i.e., cellular base-station) selection procedure used in the initial bootstrapping phase in which a device first selects a suitable base-station that allows the device to establish a connection with the core network and subsequently with the Internet. To advertise their presence, base-stations periodically broadcast information about the network in `system information` messages. Cellular devices listen to these broadcast messages and connect to a suitable base-station that satisfies the cell (re-)selection criteria based on the received signal strength of broadcast messages, cell acceptability to the device, and service type of that cell.

Unfortunately, there is no mechanism to ensure the authenticity of `system information` broadcast messages even in 5G networks currently. This allows an adversary to spoof [31, 42] or tamper with [63] `system information` messages by a fake base-station emitting signals with a higher strength than that of the nearby legitimate base-stations. After luring the cellular devices to connect to it, the fake base-station can then launch security and privacy attacks, including DNS-redirection [54], denial-of-service (DoS) [4, 40, 56, 58], location tracking [58], activity monitoring [40, 57] and bidding down [4, 57, 58] attacks. Although the latest 5G specifications (Release 15 [3]) have introduced a new

ASIA CCS '21, June 7–11, 2021, Hong Kong, Hong Kong

ACM ISBN 978-1-4503-8287-8/21/06.
https://doi.org/10.1145/3433210.3453082

cryptographic scheme for preventing the exposure of cellular device's permanent identifier in plain-text, such a mechanism does not address the root cause of the fake base-station problem, which is the absence of authentication of `system information` broadcast messages. A broadcast authentication scheme is critical for a cellular device to verify the legitimacy of the base-station to which the device initially connects, albeit currently missing primarily due to deployment challenges and backward compatibility. This paper aims to bridge this gap by proposing a practically deployable authentication mechanism for securing the initial connection bootstrapping process between cellular devices and base-stations.

While symmetric key primitives (e.g., HMAC) can efficiently provide authentication, they fail to provide public verifiability and non-repudiation in addition to the pairwise key distribution and storage hurdle. A recent study [4] by 3GPP and other efforts [47, 65, 66] have explored using certificate-based Public Key Infrastructure (PKI) or identity-based signature schemes [25, 27] to authenticate base-stations. These techniques are, however, prohibitively expensive in terms of communication overhead and computation overheads both at the signature generator and verifier sides. Although the scheme [42] by Hussain et al. has shown the viability of PKI based authentication by introducing optimizations with smaller certificates and an offline-online signature generation mechanism to reduce the signature generation time, their scheme requires a significantly higher number of expensive cryptographic operations and introduces a prohibitively long delay at the cellular device to verify the signatures and certificate chains. The communication overheads and computational delays of these signature schemes and authentication protocols will be further aggravated in 5G networks since 5G base-stations use much higher frequency radio waves (e.g., millimeter waves) to offer faster communication in exchange for significantly smaller coverage area than the base-stations for the previous generations. This induces 5G devices to switch base-stations at a much higher rate than previous generations, further adding to the authentication and signaling overhead. Due to such significant overheads, these existing proposals are, therefore, not being adopted in the latest 5G specifications or deployed by the cellular service providers.

Any candidate protocol for authenticating initial broadcast messages in 5G cellular networks must be efficient both for the signer and the verifier. The protocol must minimize the computation overhead, especially on the verifier side to preserve battery life for cellular devices without affecting the quality-of-services and the strict scheduling constraints of broadcast messages. The authentication protocol should not also add a significant communication overhead in terms of the size of certificates, signatures, and keys since additional bytes in broadcast radio packets transmitted over licensed spectrum induce additional costs to cellular service providers.

To address these challenges, we propose `Schnorr-HIBS`, a hierarchical identity-based signature scheme, using the identity-based variant of Schnorr signatures [36]. The underlying signature scheme is computationally efficient at signer and verifier, provably-secure and has significantly low communication overhead as it does not require certificates and has a small signature size. Realizing a practical broadcast authentication protocol using such IBS for cellular networks, however, requires addressing two additional challenges: ❶ IBS schemes rely on a trusted-third-party, called Private Key Generator (PKG), to generate and assign public-private keys to different entities. Due to the large number of deployed base-stations (estimated to be more than 12 million worldwide), a single PKG will not be able to handle the workload for generating and providing keys to all of them. ❷ To verify the signature, the cellular device would be required to communicate with the PKG to obtain the public-keys for signature verification. This, however, is not possible during bootstrapping because the cellular device is not yet connected to the core network. We address these challenges by allowing the creation of multiple intermediate key generators to allocate public-private keys instead of a single PKG. The PKG requires to allocate keys for these intermediate key generators which significantly reduces the workload on the system. Our protocol only requires the verifiers, i.e., cellular devices, to have the master public key of the PKG to verify a signature sent by a base-station.

Our authentication protocol introduces a new entity called core-PKG in the authentication server function in the 5G core network. The core-PKG generates public-private key-pairs for the Access and Mobility Management Function (AMF), which is the mobility anchor point in core network and controls multiple base stations. The AMF, in turn, generates public-private key-pairs for the base-stations under its control. A base-station uses its private-key to sign system information broadcast messages by following our signature generation scheme to enable cellular devices to efficiently authenticate broadcast messages.

We also optimize our protocol implementation by using elliptic-curves for generating and verifying the signatures, and pre-computing random tokens in an offline phase to further reduce the signing stage computations. We also address different operational challenges for such an authentication algorithm: ① How to securely provide the master public key, required for authentication, to cellular devices. ② How to protect against relay attacks carried out by an adversary by just re-transmitting `system information` messages from a legitimate base-station without changes. ③ How to handle roaming scenarios, i.e. when the cellular device is outside the coverage area of its service provider and has to use the network of a partner cellular service provider.

Our technical contributions: ❶ A comprehensive characterization of the attacks enabled by fake base-stations for both 5G and 4G LTE cellular networks. ❷ `Schnorr-HIBS`, a signer and verifier efficient hierarchical identity-based signature scheme, based on Schnorr signatures, and an authentication protocol based on it enabling cellular devices to authenticate base stations they connect to. ❸ An implementation of our protocol on a testbed with off-the-shelf equipment and open-source libraries and its comparison wrt computation and communication overhead with other 3GPP proposals for cellular base-stations authentication.

2 BACKGROUND

In this section, we briefly describe the architecture of the 5G cellular networks. We also introduce identity-based signatures which are the basic building block of our authentication protocol.

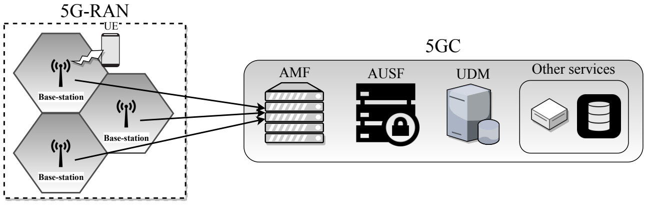


Figure 1: Cellular Network Architecture.

2.1 5G Cellular Network Architecture

A 5G cellular network can be divided into 3 main parts (see Fig. 1): User Equipment (UE), Next Generation Radio Access Network (NG-RAN) and the 5G Core Network (5GC).

UE refers to the subscriber device used to access the cellular network. The UE is provided with a Universal Subscriber Identity Module (USIM) card, provisioned by a mobile network operator. The USIM contains the permanent identity of the UE, known as Subscription Permanent Identifier (SUPI) – previously known as IMSI. In general, it is a string of 15 digits and uniquely identifies the UE globally. The UE is also provided with a temporary identity, called Globally Unique Temporary Identifier (GUTI), by the cellular network for communication with the network to avoid the exposure of the permanent identity, that is, the SUPI.

NG-RAN consists of base-stations that UEs can connect to using radio transmission. 5G cellular networks allow both 5G base-stations and 4G LTE base-stations. The base-stations broadcast `system information` messages, including a Master Information Block (MIB) and multiple System Information Block (SIB) messages, at regular intervals. The MIB message is broadcast every 80 ms (with network dependent periodic repetitions in these 80 ms) and the SIB1 message is broadcast every 160 ms (with network dependent periodic repetitions in these 160 ms). The other SIB messages can either be broadcast periodically or upon request by the UE. MIB and SIB1 together are referred to as *minimum SI* as they are the most important messages to enable further communication between the UE and the base-stations. The UE listens for SI messages and connects to the base-station with the highest signal strength.

5GC is the brain of the 5G cellular network and houses several components to provide services to the UEs. An important component is the Access and Mobility Management Function (AMF), known as Mobility Management Entity (MME) in 4G LTE. The AMF supports UE authentication, mobility management and paging, handles the NAS layer traffic and security, and checks UE's roaming rights. The AMF authenticates the UE in collaboration with the Authentication Server Function (AUSF) and Unified Data Management (UDM).

2.2 Identity-Based Signatures

A Public-Key Infrastructure (PKI) is typically used to manage the public-keys required for digital signatures and to bind public-keys to their owning entities. In a traditional Certification Authority (CA) based PKI, certificates are used to encode the public-keys, information about their owning entities, their validity periods etc. along with a signature by the CA to prove their legitimacy. Identity-based cryptography, on the other hand, eliminates the need for certificates and instead relies on a Private Key Generator (PKG) to generate private keys from a master secret and distribute them to the participating entities. This PKG can also be used to enable identity-based signatures. In what follows, we describe the cryptographic primitives required for our signature scheme.

Notation. Given two primes p and q, we define a finite field $\mathbb{F}_p$ and a group $\mathbb{Z}_q$. We work on $E(\mathbb{F}_p)$ as an elliptic curve (EC) over $\mathbb{F}_p$, where $P \in E(\mathbb{F}_p)$ is the generator of the points on the curve. We denote a scalar and a point on a curve with small and capital letters, respectively. $x \xleftarrow{\$} S$ denotes a random uniform selection of x from a set S. $||$ denotes string concatenation. EC scalar multiplication is denoted as $x \times P$, and all EC operations use an additive notation. $\vec{x}_l$ denotes a vector of dimension l, i.e., $\vec{x}_l = \{x_1, \dots, x_l\}$. We define two hash functions $\mathsf{H}_1\colon \{0,1\}^{l_1} \to \mathbb{Z}_q$ and $\mathsf{H}_2\colon \{0,1\}^* \to \mathbb{Z}_q$, where l_1 denotes our user identity space. We view these hash functions as random oracles in our security analysis [22].

Hierarchical Identity-Based Signatures. Hierarchical identity-based cryptography [37] is a generalization of identity-based cryptography that delegates trust like an organizational hierarchy. The key generating entities (i.e., PKGs) are arranged in a tree structure and the identity of a user at depth k is a vector of dimension k, i.e., $\vec{ID}_k = \{ID_1, \dots, ID_k\}$. We ignore the subscript (e.g., k) if the depth is not important. A hierarchical identity-based signature (HIBS) has four algorithms similar to those in identity-based signature, except that the key generation algorithm (i.e., `Extract`) is used to generate private keys for a given user who is either a normal user in the systems or a lower-level PKG. We formally define the notion of HIBS in the following.

Definition 2.1. A hierarchical identity-based signature scheme is defined by four algorithms $\mathtt{HIBS} = \{\mathtt{Setup}, \mathtt{Extract}, \mathtt{Sig}, \mathtt{Ver}\}$.

$\underline{(sk_{ID_0}, params) \leftarrow \mathtt{HIBS.Setup}(1^\kappa)}$: Given the security parameter κ, the PKG selects master secret key sk_{ID_0}, computes master public key mpk and system parameters $params$ (an implicit input to all the following algorithms).

$\underline{(sk_{ID_k}, \vec{Q}_{ID_k}) \leftarrow \mathtt{HIBS.Extract}(\vec{ID}_k, sk_{ID_{k-1}}, \vec{Q}_{ID_{k-1}})}$: Given identity vector $\vec{ID}_k = (ID_1, \dots, ID_k)$ at level k, and the commitment values $\vec{Q}_{ID_{k-1}}$ the private key $sk_{ID_{k-1}}$ of the entity at depth $k-1$, it outputs the private key for the entity ID_k and its corresponding commitment value vector $\vec{Q}_{ID_k} = (Q_{ID_1}, \dots, Q_{ID_k})$.

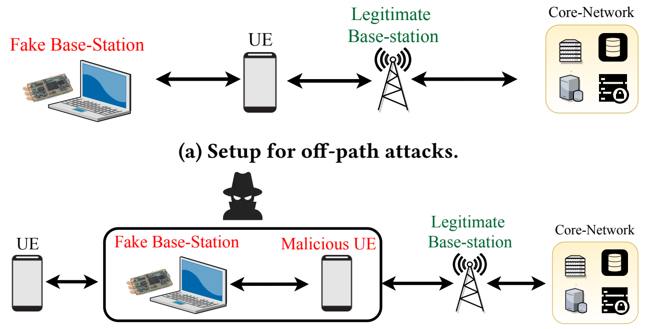
$\underline{\sigma \leftarrow \mathtt{HIBS.Sign}(m, sk_{ID})}$: Given a message m and sk_{ID}, returns a signature σ.

$\underline{d \leftarrow \mathtt{HIBS.Verify}(m, \vec{ID}_k, \vec{Q}_{ID_k}, \sigma)}$: Given m, σ and $(\vec{ID}_k, \vec{Q}_{ID_k})$ as input, if the signature is valid, it returns $d = 1$, else $d = 0$.

Existential Unforgeability. Following [28], we define the notion of existential unforgeability for selective-ID, adaptive chosen message-and-identity attack (EF-sID-CMIA) for a HIBS scheme in the following game between a challenger C and adversary $\mathcal{A}$. After the initialization phase, $\mathcal{A}$ has access to the following oracles. (i) Key extraction oracle O_{Ex}: Given the user identity ID, output the private key of the user sk_{ID}. (ii) Sign oracle O_{Sign}: Given the user identity ID and a message m, output a valid signature σ.

Definition 2.2. The EF-sID-CMIA experiment for a HIBS scheme is defined as follow.

- C runs $\mathtt{HIBS.Setup}(1^\kappa)$ and returns mpk and $params$ to $\mathcal{A}$.
- $(\vec{ID}^*, \vec{Q}_{ID^*}, m^*, \sigma^*) \leftarrow \mathcal{A}^{O_{Ex}, O_{Sign}}(mpk, params)$



(a) Setup for off-path attacks.

(b) Setup for man-in-the-middle attacks.

Figure 2: Common fake base-station configurations for carrying out attacks.

$\mathcal{A}$ wins the above experiment if `HIBS.Verify`$(m^*, \vec{ID}^*, \vec{Q}_{ID^*}, \sigma^*)$ returns 1 and $\vec{ID}^*$ and any prefix of $\vec{ID}^*$ was not queried to O_{Ex}. Additionally, no query on $(\vec{ID}', m^*)$ where $\vec{ID}'$ is a prefix of $\vec{ID}^*$ should have been made to O_{Sign} during the above game.

3 CHARACTERIZATION OF ATTACKS ENABLED BY FAKE BASE STATION

Fake base-stations have been shown to be realizable in practice [46, 52, 58, 60] using off-the-shelf hardware and open source cellular software stacks. To make the fake base-station work, the attacker-controlled radio transmits messages at a higher signal strength than the legitimate base-stations to force users to connect to it over legitimate ones. Fig. 2a shows a fake base-station setup for carrying out off-path attacks whereas Fig. 2b shows the setup for Man-in-the-Middle (MitM) relay attacks.

To protect against fake base-station attacks, improvements have been introduced in different versions of cellular protocol specifications, including 3G, 4G LTE, and 5G. These defenses, however, do not address the root cause of attacks, i.e., the lack of authentication for base-stations during connection bootstrapping and, thus, still enable attacks even in 5G networks. To better understand the causes and implications of these attacks, we now first analyze the attacks enabled by fake base-stations in 4G networks, and then whether these attacks are still unaddressed in the latest release of the 5G 3GPP protocol specifications (Release-15 [3]). See Table 1 for a summary of these attacks.

DoS attacks. Fake base-stations can carry out different DoS attacks against UEs already connected to them [4, 40, 56, 58]. This is enabled by the existence of several control plane messages that have no integrity protection. A fake base-station can send reject messages (`auth_reject`, `RRC_Reject`, `NAS_Reject`) to UEs connected to the network and force them to disconnect from the network causing a DoS attack. A fake base-station can also deny some or all network services to a UE by tampering with the `Tracking Area Update` (TAU) procedure [56, 58]. The location update, i.e., the tracking area update (TAU) procedure is used by the UE to notify the network of its current tracking area and can be used by the network to deny some services to the UEs based on the capabilities of the UE or the serving network. The fake base-station can also force the UE to deem the SIM card invalid for the network and cause a persistent DoS until the UE is rebooted. The tracking area update procedure procedure is not available in 5G networks, but this attack can still be carried out by forcing the UE to user older cellular protocols via bidding-down attacks.

Location tracking. During the initial `attach procedure` in 4G networks, a UE sends either its permanent identifier IMSI or the temporary identifier GUTI in plain-text for device identification. The UE can also be explicitly asked to send its IMSI in plain-text when the network sends an `Identity Request` message, which does not have any integrity protection. Exploiting this lack of authentication, a fake base-station can send illegitimate `Identity Request` message and capture victim's IMSI. Such an attack is called IMSI-catching [49, 51]. It allows the attacker to obtain IMSIs for nearby victims and then use them for tracking the victims movements by exploiting vulnerabilities in the paging protocol [41]. The temporary identifier GUTI can also be used for tracking victims in a similar way as many network operators do not refresh the GUTI regularly because of the absence of clear guidelines by 3GPP specifications [46]. The 5G protocol addressed such issue by requiring the UE to encrypt the SUPI (IMSI) using a randomized public key encryption scheme. This prevents attackers from getting information about the SUPI and, thus, prevents them from being able to track their victims using it. Also, the 5G specifications now explicitly require the network operators to regularly refresh GUTIs to prevent attackers from tracking victims for longer time periods. However, these attacks are still possible by first carrying out other DoS or bidding-down attacks to downgrade the connection protocol to 4G LTE or lower. The `Measurement Report` message also has been used by attackers to detect a UE precise location [58]; this message is now confidentiality protected in 5G.

Bidding down attacks. These attacks allow a fake base-station to force the UEs to use older cellular protocols (e.g., 2G, 3G). These protocols often provide a lower service quality, and the confidentiality and integrity protections they provide are generally susceptible to being broken. During the TAU procedure the fake base-station can send a `TAU_Reject` message with the cause number 7 ("LTE services not allowed") as a reply to the `TAU_request` message [58]. The UE then starts to search for 2G and 3G networks in the area, which are susceptible to many more attacks. Such an effect can also be achieved using `Service_Reject` and `Attach_Reject` messages as they lack integrity protection too.

Traffic monitoring. UEs can be forced to connect to the fake base-station using weaker cryptographic protections of the older cellular protocols [50] by using bidding-down attacks. This might allow adversaries to monitor call and data traffic for the UE by breaking the cryptographic protocols used for providing confidentiality protections to UEs [35]. Another privacy attack on the 4G LTE Authentication and Key Agreement (AKA) protocol and the enhanced 5G-AKA protocol allows an attacker to learn the activity patterns of subscribers (e.g., number of calls and SMSs sent at a given time) and monitor those patterns remotely over time [57]. Other attacks can monitor UE's internet usage or make the UEs connect to malicious websites by redirecting the UE's DNS queries to an attacker controlled DNS server [54]. An authentication relay attack [40] allows an attacker's malicious UE to impersonate a legitimate UE and

Attack	Attack Category	4G LTE	5G	Impact
Send Service_Reject or Attach_Reject [58]	DoS	Yes	Yes	Denial of all services;
Send Authentication_Reject [40]	DoS	Yes	Yes	Denial of all services;
Replay RRC_Resume_Request [4]	DoS	N/A	Yes	Denial of services
Manipulate Self Organizing Networks (SON) [56]	DoS	Yes	Yes	Call dropping; Increase in power consumption for UE; Increased handovers and signaling load; Legitimate base-station blacklisted
Send RRC_Reject or NAS_Reject [4]	Bidding-down; DoS	Yes	Yes	Denial of all services; Downgrade to 2G/3G/4G LTE
Send TAU_Reject [58]	Bidding-down; DoS	Yes	N/A	Denial of some or all services; Downgrade to 2G/3G/4G LTE
Modify UE_Capability_Information [57]	Bidding-down	Yes	Yes	Denial of some services; Lower data rate; Downgrade to 2G/3G/4G LTE
Authentication Relay Attack [40]	Activity monitoring; DoS	Yes	Yes	Complete or selective DoS; Location history poisoning; Network profiling
Attack on 5G AKA protocol [57]	Activity monitoring	Yes	Yes	Remotely monitor subscriber activity patterns like calls, SMS etc.
Check Measurement_Report and UE_Information messages [58]	Location tracking	Yes	No	Fine-grained location reveal
Modify Attach_Request or TAU_Request [57]	Power Drain	Yes	N/A	High power drain in IoT devices

Table 1: Attacks enabled by fake base-stations in 4G LTE and 5G cellular networks. N/A means the particular protocol interaction does not exist in the corresponding cellular protocol.

poison the location history or profile the network usage of the legitimate UE. 5G raises the bar for this attack by encrypting the SUPI (IMSI) instead of sending it in plain-text, but such an attack is still possible in conjunction with other physical layer vulnerabilities.

4 OVERVIEW OF OUR SOLUTION

Adversary Model. We consider a Dolev-Yao adversary model [32] in which the adversary can drop, modify, inject or eavesdrop messages sent by legitimate participants over the public radio channel. According to this model, the adversary is capable of setting up fake base-stations and emitting unauthenticated broadcast messages with a higher signal strength than the legitimate base-stations. The adversary cannot, however, physically access and tamper the legitimate base-stations, cellular devices or core-network components, and cannot access the secret keys or other sensitive information stored in a target cellular device's USIM or base-stations.

Scope of our Solution. Our solution allows cellular devices to reliably authenticate a base-station before establishing a connection by ensuring the authenticity of the public broadcast messages. We do not consider passive attacks caused by adversaries eavesdropping the traffic between the target device and legitimate base-stations over the public radio channel. We also do not consider DoS attacks using a wireless jammer operating at the physical layer. Finally, our solution is envisioned for 5G cellular networks, but can be extended to 4G LTE, 3G and 2G networks with minimal modifications.

Our Authentication Protocol. Our protocol allows a UE to verify the identity of the base-station it is connecting to and validate the system information messages being sent by the base-station. Our protocol is based on a HIBS scheme (details in Section 5) and organized according to a 3-layered system consisting of: ❶ the core-PKG (hosted by the 5GC), ❷ AMFs, and ❸ base-stations. The core-PKG is co-located with the Authentication Server Function (AUSF) in the core-network and is in charge of generating the public-private key pairs for all the AMFs deployed for a particular network operator. We provide a high-level overview of our authentication protocol below (see Section 6 for further details).

Core-PKG generates its public-private key pair $[sk_{PKG}, PK_{PKG}]$ during the initial setup phase. The PK_{PKG} is installed inside the USIM of all the UEs for that particular network operator during their registration. The AMFs periodically send a key generation request to their network operator's core-PKG. They send their AMF_ID to the core-PKG and receive a public-private key pair $[sk_{AMF}, PK_{AMF}]$ and ID_{AMF} from the core-PKG. The ID_{AMF} is a concatenation of AMF_ID and the expiration timestamp of the particular key-pair. Similarly, the base-stations send a key generation request to the AMF serving their particular tracking area. The base-stations send their NRCell_ID to the AMF and receive a public-private key pair $[sk_{BS}, PK_{BS}]$ and ID_{BS}. The ID_{BS} is a concatenation of NRCell_ID and the expiration timestamp of the particular key-pair.

The base-stations use their assigned private key sk_{BS} to sign the system information broadcast messages using the Schnorr-HIBS (Section 5) and generate a signature sig_{SIB1}. The base-stations attach sig_{SI} to the system information message along with ID_{BS}, PK_{BS}, ID_{AMF}, PK_{AMF}. The UE uses this information to verify sig_{SI}. The UE first verifies that the keys PK_{BS} and PK_{AMF} have not expired by checking the expiry timestamps embedded in the ID_{BS} and ID_{AMF}. If the timestamps have not expired, the UE then verifies the signature of the system information message. If this verification step is successful, the UE connects to the base-station.

Algorithm 1: Schnorr-HIBS

```
$(sk_{ID_0}, params) \leftarrow \texttt{HIBS.Setup}(1^{\kappa})$: This algorithm is run once by the PKG to setup the system.
    (1) Select primes $p$ and $q$
    (2) $x \overset{\$}{\leftarrow} \mathbb{Z}_q$ and $mpk \leftarrow x \times P \bmod p$
    (3) Set $sk_{ID_0} \leftarrow x$ and $params \leftarrow \{p, q, \mathsf{H}_1, \mathsf{H}_2\}$

$(sk_{ID_k}, \vec{Q}_{ID_k}) \leftarrow \texttt{HIBS.Extract}(\vec{ID}_k, \vec{Q}_{ID_{k-1}}, sk_{ID_{k-1}})$: This algorithm is run by the user at level $k-1$ to generate the private-key for the user at level $k$.
    (1) $b \overset{\$}{\leftarrow} \mathbb{Z}_q, Q_{ID_k} \leftarrow b \times P \bmod p$
    (2) $\vec{Q}_{ID_k} \leftarrow (\vec{Q}_{ID_{k-1}}, Q_{ID_k}), c_{ID_k} \leftarrow \mathsf{H}_1(ID_k || \vec{Q}_{ID_k})$
    (3) $sk_{ID_k} \leftarrow sk_{ID_{k-1}} \cdot c_{ID_k} + b \bmod q$

$\sigma \leftarrow \texttt{HIBS.Sign}(m, sk_{ID_k})$: This algorithm is run by the user $(ID_k, Q_{ID_k})$ to generate a signature $\sigma$ on message $m$.
    (1) $r \overset{\$}{\leftarrow} \mathbb{Z}_q, R \leftarrow r \times P \bmod p, h \leftarrow \mathsf{H}_2(m||R)$
    (2) $s \leftarrow sk_{ID_k} \cdot h + r \bmod q$
    (3) $(\sigma = \langle s, h \rangle, Q_{ID_k})$

$d \leftarrow \texttt{HIBS.Verify}(m, \vec{ID}_k, \vec{Q}_{ID_k}, \sigma)$
    (1) $c_{ID_i} \leftarrow \mathsf{H}_1(ID_k || \vec{Q}_{ID_i})$ for $i = \{1, \ldots, k\}$
    (2) $Q \leftarrow \sum_{j=1}^{j=k-1} (\prod_{i=j+1}^{k} c_{ID_i}) \times Q_{ID_j} \bmod p$
    (3) $R^{'} \leftarrow s \times P - h \left( \prod_{i=1}^{k} (c_{ID_i}) \times mpk + Q + Q_{ID_k} \right) \bmod p$
    (4) **If** $h = \mathsf{H}_2(m||R^{'}), d \leftarrow 1$ , **else** $d \leftarrow 0$.
```

5 SCHNORR-HIBS

In this section, we details our hierarchical identity-based signature scheme, called Schnorr-HIBS, and discuss its security.

5.1 Proposed Scheme

Our signature scheme is a based on the identity-based signature scheme proposed by Galindo and Garcia [36] defined using Schnorr signatures [55]. We describe our signature scheme in Algorithm 1.

The user identity in our scheme is considered to be an identity vector. More precisely, $\vec{ID}_k$ is a vector which includes the identities of all the entities leading to up to ID_k, from the node directly following the root (i.e., $\vec{ID}_k = (ID_1, \ldots, ID_k)$). The PKG, which is also regarded as the user at level 0, selects its private-key sk_{ID_0}, and publishes the system-wide parameters $params$. Our main departure from the scheme in [36] is our `HIBS.Extract` algorithm, where in addition to the PKG, users can also generate keys for the lower level users. The private-key is generated by computing a Schnorr signature [55] on $\vec{ID}_k$ and all the corresponding commitment values $\vec{Q}_k$. The `HIBS.Sign` algorithm in our scheme is a Schnorr signature signing algorithm. For practicality purposes, we can assume that Q_{ID_k} is only sent to the verifier once, or can be downloaded from a public bulletin. Our verification algorithm is a slight variation of that in [36], where we need the public-key vector to verify the signature. As depicted in Steps 2 and 3 of our `Verify` algorithm, the cost of our verification algorithm is linear with the depth of the tree in our hierarchical organization.

5.2 Security Analysis

Like our scheme, the scheme in [36] uses Schnorr signatures in its extract and signature generation steps. The difference in our scheme is the extension of the extract algorithm to allow users in the system to generate private-keys for the users directly below them in the organization hierarchy. Therefore, like [26, 36] by relying on variations of the forking lemma [21, 24], we can achieve a non-tight security for our scheme based on the hardness of the elliptic curve discrete logarithm problem (ECDLP). The following theorem provides a sketch proof of the security of our scheme based on [26].

THEOREM 5.1. *If an adversary $\mathcal{A}$ can $(\epsilon, q_{Ex}, q_S, q_{\mathsf{H}_1}, q_{\mathsf{H}_2})$-break the scheme proposed in Algorithm 1, in the sense of Definition 2.2 in the random oracle model, where $q_{Ex}, q_S, q_{\mathsf{H}1}$, and $q_{\mathsf{H}2}$ are the queries to O_{Ex}, O_{Sign} and hash functions H_1 and H_2, then one can construct another algorithm C to break DLP via either of the following reduction algorithms:*

1. *Algorithm $\mathcal{R}_1$ that $\epsilon_{\mathcal{R}_1}$-breaks ECDLP where $\epsilon_{\mathcal{R}_1} \geq \frac{\epsilon^2}{\mathbf{e}\, q_{Ex}\, q_{\mathsf{H}_1}}$*
2. *Algorithm $\mathcal{R}_2$ that $\epsilon_{\mathcal{R}_2}$-breaks ECDLP where $\epsilon_{\mathcal{R}_2} \geq \epsilon \left(\frac{\epsilon}{(q_{\mathsf{H}_1} + q_{\mathsf{H}_2})^2} - \frac{1}{q} \right)$*
3. *Algorithm $\mathcal{R}_3$ that $\epsilon_{\mathcal{R}_3}$-breaks ECDLP where $\epsilon_{\mathcal{R}_3} \geq \epsilon \left(\frac{\epsilon^3}{(q_{\mathsf{H}_1} + q_{\mathsf{H}_2})^6} - \frac{3}{q} \right)$,*

where $\mathbf{e}$ is the base of natural logarithm. Please refer to Appendix A for proof.

6 INSTANTIATION OF SCHNORR-HIBS FOR 5G NETWORKS

We instantiate the authentication protocol for 5G cellular networks based on the Schnorr-HIBS scheme (see Section 5) with levels $k = 2$. The Core-PKG located in the Authentication Server Function (AUSF) in the core network serves as the PKG at level 0, the AMFs are placed at level 1, and the base-stations are placed at level 2. In this section, we outline the detailed design of our protocol and the rationale behind various design decisions.

6.1 Design Decisions

Hierarchical architecture. We specify a 2-tier hierarchical architecture for our protocol: the core-PKG generates the keys for the AMFs and the AMFs generate the signing keys for the base-stations. We use a hierarchical approach over a flat approach where a core-PKG generates keys for all base-stations for several reasons: ❶ A single PKG for all base-stations introduces a single point of failure. If this component goes down or the private keys are leaked, the whole cellular network would be vulnerable. ❷ Generating keys for the large number of base-stations (estimated to be more than 12 million worldwide) at a single place introduces a huge computational overhead. Such overhead increases if the key refresh interval for base-stations is very low. For a base-station with key expiry_time set at 1-hour, 10-minutes and 1-minute, the PKG has to generate 24, 144 and 1440 keys per day, respectively. For 100,000 base-stations this value would be 2.4 million, 14.4 million and 144 million keys daily. ❸ Having the PKG at a centralized location adds significant communication latency to the key generation operation, due to the physical distance between the PKG and base-stations. For instance, if the PKG and a base station are ~3000 miles (=4828.032 km) apart

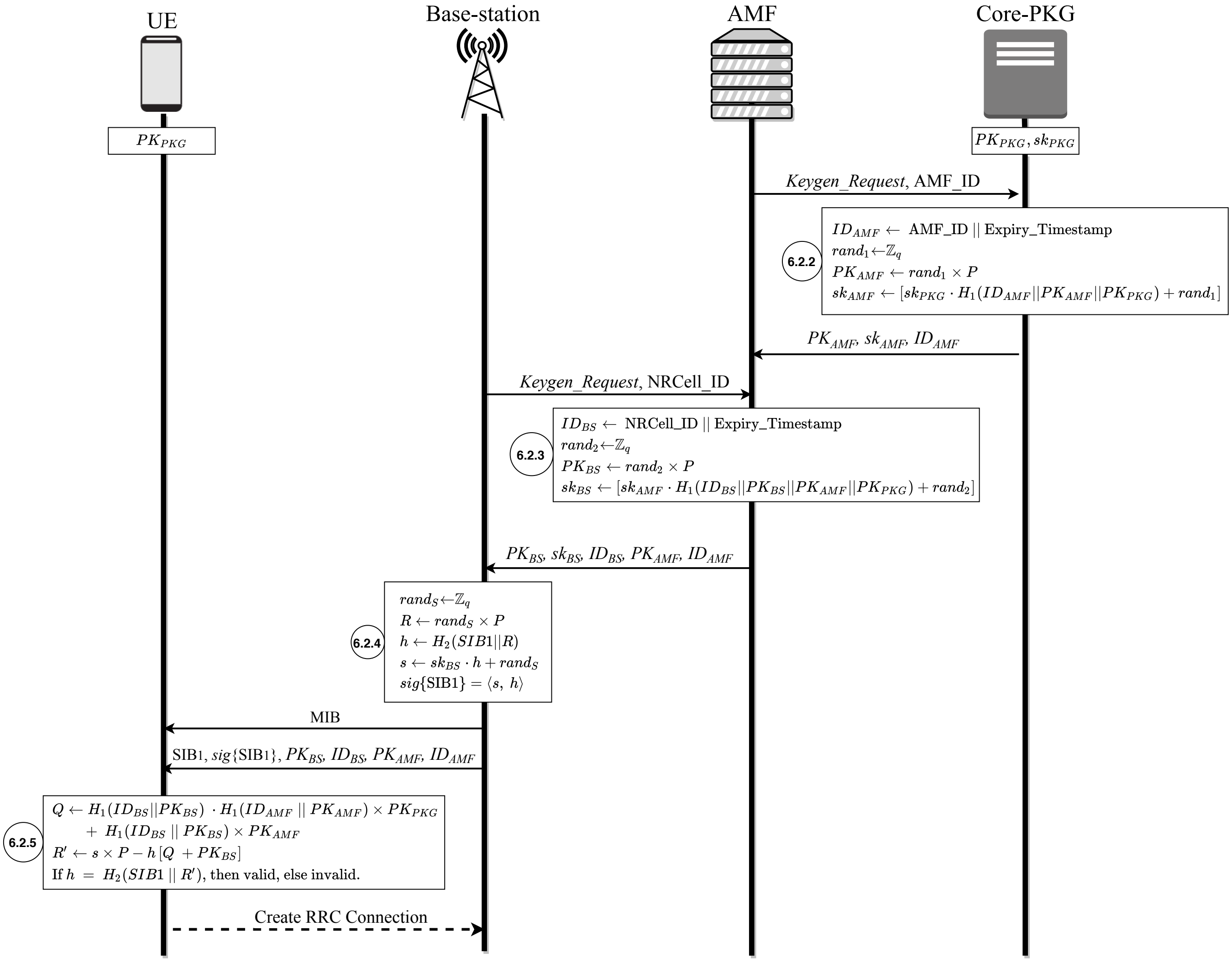

UE
Base-station
AMF
Core-PKG
PK_{PKG}
PK_{PKG}, sk_{PKG}
Keygen_Request, AMF_ID
6.2.2
$ID_{AMF} \leftarrow$ AMF_ID || Expiry_Timestamp
$rand_1 \leftarrow \mathbb{Z}_q$
$PK_{AMF} \leftarrow rand_1 \times P$
$sk_{AMF} \leftarrow [sk_{PKG} \cdot H_1(ID_{AMF}||PK_{AMF}||PK_{PKG}) + rand_1]$
$PK_{AMF}, sk_{AMF}, ID_{AMF}$
Keygen_Request, NRCell_ID
6.2.3
$ID_{BS} \leftarrow$ NRCell_ID || Expiry_Timestamp
$rand_2 \leftarrow \mathbb{Z}_q$
$PK_{BS} \leftarrow rand_2 \times P$
$sk_{BS} \leftarrow [sk_{AMF} \cdot H_1(ID_{BS}||PK_{BS}||PK_{AMF}||PK_{PKG}) + rand_2]$
$PK_{BS}, sk_{BS}, ID_{BS}, PK_{AMF}, ID_{AMF}$
6.2.4
$rand_S \leftarrow \mathbb{Z}_q$
$R \leftarrow rand_S \times P$
$h \leftarrow H_2(SIB1||R)$
$s \leftarrow sk_{BS} \cdot h + rand_S$
$sig\{\text{SIB1}\} = \langle s,\ h \rangle$
MIB
SIB1, $sig\{\text{SIB1}\}$, $PK_{BS}, ID_{BS}, PK_{AMF}, ID_{AMF}$
6.2.5
$Q \leftarrow H_1(ID_{BS}||PK_{BS}) \cdot H_1(ID_{AMF} \,||\, PK_{AMF}) \times PK_{PKG} + H_1(ID_{BS} \,||\, PK_{BS}) \times PK_{AMF}$
$R' \leftarrow s \times P - h\,[Q + PK_{BS}]$
If $h = H_2(SIB1 \,||\, R')$, then valid, else invalid.
Create RRC Connection

Figure 3: Our protocol for authenticating 5G cellular base-stations.

geographically, i.e., one entity in New York and the other one in Los Angeles, the average round-trip delay would be approximately 68.42ms [13], which is very high compared to our hierarchical scheme's key generation time which is 0.03 ms only.

Choice of messages to sign. System information messages are broadcast periodically by the base-stations to allow UEs to initiate a connection to them. System information messages are divided into a master information block (MIB) and multiple system information block (SIB) messages [8]. MIB includes the basic parameters required by the UE to acquire the SIB1 message. The SIB1 message is the most important system information message, and contains the base-station selection parameters, scheduling info for the rest of the SIB messages, whether one or more SIB messages are only provided on-demand and configuration needed by the UE to perform the system information request. All messages transmitted by the base-stations can be integrity protected using our signature scheme; however, signing all such messages would add a significant communication and computation overhead. To provide a way for the UEs to verify whether a base-station is legitimate or not, our protocol only requires the base-stations to sign the SIB1 message. This signature is used to authenticate the base-stations.

Construction of Identities. Our protocol requires assigning IDs to the AMFs and base-stations. We utilize the IDs for the dual purpose of uniquely identifying the AMFs/base-stations as well as for communicating the validity period of their signing keys. For ID_{AMF} we use a concatenation of AMF-Identifier (AMF_ID) [5] and an expiry timestamp. AMF_ID comprises of the AMF's region ID, set ID, and an AMF pointer. AMF_ID is a bit-string of size 24 bits and uniquely identifies the AMF for a particular network operator. For ID_{BS} we use a concatenation of NRCell_ID [5] and an expiry timestamp. NRCell_ID is a string of size 36 bits and uniquely identifies a base-station for a particular mobile network operator. Each expiry timestamp is 32 bits long. Therefore, the ID_{AMF} can be a maximum of 7 bytes and ID_{BS} can be a maximum of 9 bytes.

Validity period of the keys. Instead of using complex key revocation techniques, we assign different validity periods to each generated key-pair after which the keys would need to be refreshed. For the core-PKG, we create the key-pair with a 1 year validity period by default as it needs to be installed inside the UE's USIM, and requires a confidentiality and integrity protected channel to be updated. The core-PKG needs to be physically secured and protected so that its private key is not leaked. The AMF key-pair on the other hand has a default validity period of 24 hours. AMFs are located in the 5G core-network either in a virtualized deployment or in a physically secure location so that they are not tampered with. Even if the AMF's private key is compromised, the adversary would only be able to generate keys for its malicious base-stations for 1 day, after which the keys will expire. For the base-stations, we generate a key-pair valid for only 10 minutes. Base-stations are located around the world in physically insecure areas. Therefore, it may be easier for the attacker to compromise them. A validity period of only 10 minutes minimizes the period during which an attacker can launch attacks, even if it obtains a base-station's private key. These validity periods are recorded in the expiry timestamps in the ID_{AMF} and ID_{BS} for AMF and base-stations, respectively, as well as in the UE's USIM for the core-PKG. These are the default validity periods and can be changed by the network operators when required. Since our key generation is efficient (1000 keys per 29-msec), its impact on Core-PKG or AMF is negligible.

6.2 Protocol Description

We now detail our authentication protocol steps. We abstract some cryptographic details for readability. For instance, we do not explicitly mention the mod operation, but all the operations in $E(\mathbb{F}_p)$ are executed in mod p and operations in $\mathbb{Z}_q$ are in mod q. Figure 3 gives a graphical representation of our protocol for a 5G scenario. A formal proof of our authentication protocol using the automated cryptographic protocol verifier - ProVerif is shown in Appendix B.

6.2.1 Initialization phase for the core-PKG. The core-PKG generates the public system parameters and its own public-private key-pair during the initialization phase. This phase is executed in the beginning of the 5GC deployment. The default validity period of the core-PKG's keys is 1 year. The public key of the core-PKG along with its expiry date is installed in the USIM of all UEs during initial registration. The core-PKG's public key installed in the USIM has to be replaced, whenever the core-KGC refreshes its keys. This can be done using the confidentiality and integrity protected channel created between the AMF and the UE after mutual authentication. The core-PKG uses the `Setup` step from Algorithm 1 to generate its key-pair $\{sk_{PKG},\ PK_{PKG}\}$.

$$sk_{PKG} \leftarrow \mathbb{Z}_q,\ PK_{PKG} \leftarrow sk_{PKG} \times P$$

6.2.2 Key generation for the AMF. AMFs send, through a secure channel, a key generation request to the core-PKG with its `AMF_ID`. To generate the AMF's key-pair, the core-PKG first creates an ID_{AMF} by concatenating the `AMF_ID` and an expiry timestamp for the key being generated. The core-PKG then generates a public-private key-pair $[sk_{AMF},\ PK_{AMF}]$ by using the `Extract` step from Algorithm 1. The AMFs have to periodically refresh their key-pair by sending the key-generation request to the core-PKG, when nearing the key-pair expiration.

$$rand_1 \xleftarrow{\$} \mathbb{Z}_q,\ PK_{AMF} \leftarrow rand_1 \times P$$
$$sk_{AMF} \leftarrow [sk_{PKG} \cdot \mathsf{H}_1(ID_{AMF}||PK_{AMF}||PK_{PKG}) + rand_1]$$

6.2.3 Key generation for the base-station. Base-stations send a key generation request to the AMF serving their particular tracking area. In case of multiple AMFs serving their tracking area, the base-stations can choose to send the key generation request to one or multiple AMFs and keep any one set of public-private keys and discard the others or keep all the pairs till they expire. The base-stations send their identifier `NRCell_ID` to the selected AMF(s) with the key-generation request and receive a public-private key-pair $[sk_{BS},\ PK_{BS}]$. They also receive ID_{BS}, ID_{AMF} and PK_{AMF} back. The ID_{BS} is a concatenation of the base-station's `NRCell_ID` and the expiration timestamp of the key-pair. The base-stations verify that the AMF's keys are valid by checking the expiry timestamp included with ID_{AMF}. Like the AMFs, the base-stations have to ensure that they get their key-pairs refreshed before their keys or their serving AMF's keys expire.

$$rand_2 \xleftarrow{\$} \mathbb{Z}_q,\ PK_{BS} \leftarrow rand_2 \times P$$
$$sk_{BS} \leftarrow [sk_{AMF} \cdot \mathsf{H}_1(ID_{BS}||PK_{BS}||PK_{AMF}||PK_{PKG}) + rand_2]$$

6.2.4 Signing phase at the base-station. The base-stations sign the SIB1 message via `Sign` step of Algorithm 1 and generate the signature sig_{SIB1}. They attach the sig_{SIB1}, ID_{BS}, PK_{BS}, ID_{AMF} and PK_{AMF} along with the SIB1 message broadcast. Before signing, the base-station needs to ensure that their own keys and their serving AMF have not expired.

$$rand_S \xleftarrow{\$} \mathbb{Z}_q,\ R \leftarrow rand_S \times P$$
$$h \leftarrow \mathsf{H}_2(SIB1||R)$$
$$s \leftarrow sk_{BS} \cdot h + rand_S$$

where $\langle s,\ h \rangle$ is the signature.

6.2.5 Verification phase at the UE. The UE uses the IDs and the PKs sent by the base-station attached to the SIB1 message to verify sig_{SIB1}. The UE first verifies that the keys PK_{BS} and PK_{AMF} are not expired by looking at the expiry timestamps embedded in the ID_{BS} and ID_{AMF}. If the timestamps have not expired, the UE then verifies the signature sig_{SIB1}. For verification, the UE uses the public keys of core-PKG, AMF and the base-station:

$$Q \leftarrow \mathsf{H}_1(ID_{BS}||PK_{BS})\ \cdot \mathsf{H}_1(ID_{AMF}\ ||\ PK_{AMF}) \times PK_{PKG}$$
$$+\ \mathsf{H}_1(ID_{BS}\ ||\ PK_{BS}) \times PK_{AMF}$$
$$R' \leftarrow s \times P - h\,[Q\ + PK_{BS}]$$
$$\text{If } h\ =\ \mathsf{H}_2(SIB1\ ||\ R'), \text{ then valid, else invalid.}$$

Authentication failure action: In case of authentication failure or the absence of authentication capabilities at the base-station, the UE does not connect to the base-station and keeps searching for other base-stations available in the area. If there are no available base-stations that can be authenticated, the UE can connect to an unauthenticated base-station or keep looking for a base-station that can be authenticated. We propose this to be a UE specific choice, which can be configured depending on the mobile user's security/connectivity needs. If the UE decides to connect to an unauthenticated base-station, it keeps checking the `system information` messages to find a base-station that can be authenticated.

6.3 Other Optimizations

High performance elliptic-curves. We use the FourQ [29] elliptic-curves for our protocol implementation. FourQ is an EC that is defined by the complete twisted Edwards equation [23] $\mathcal{E}/F_{p^2}$: $-x^2 + y^2 = 1 + dx^2y^2$. FourQ is one of the fastest elliptic curves that provides 128-bit security level. FourQ also offers really fast EC addition, which is really helpful for our scheme.

Pre-computation of random tokens. The Signing phase of the `Schnorr-HIBS` signature scheme (Algorithm 1) requires a random token to be generated and then multiplied with the generator P. This is an EC-point multiplication and is time consuming. To reduce computation during the signing stage a large batch of random tokens can be generated in the offline phase and used when they are actually required for signing. This optimization substantially reduces the signing phase computation cost. We call this variant `Schnorr-HIBS-P`.

6.4 Handling Roaming Scenario

Roaming service enables a UE to maintain cellular connectivity even when outside the coverage area of its primary network operator. In this scenario, the UE can be serviced by another network operator with an existing agreement with the UE's primary service provider. The current network operator in this case is called the *Serving Network* and the primary service provider is called the *Home Network*.

Our protocol requires storing the public-key of the core-PKG (deployed by a particular network operator) in the UE's USIM. This public-key allows the UEs to authenticate all the base-stations managed by their own network operator. However, this would not work when roaming. To authenticate the base-stations using our protocol while roaming, the UE requires the public-key of the core-PKG of the network operator hosting the serving network. During roaming, the UE will need to request the public-key of the serving network's core-PKG signed by the UE's home network's core-PKG, using non-3GPP radio access networks (e.g., Wi-Fi). eSIM can facilitate this process since it allows remote certificate provisioning without compromising security.

6.5 Protection Against Relay Attacks

Our authentication protocol provides protections against fake base-stations by allowing the UE to authenticate the `system information` messages. There is still, however, the possibility of a relay attack where an adversary listens to the `system information` messages broadcast by legitimate base-stations and re-transmits them to UEs without alteration with a higher signal strength. This relay attack causes the UEs to mistake the fake base-station for a legitimate base-station and connect to it. Distance-bounding protocols have been proposed to protect against these relay attacks [33, 53, 61]. Implementing such protocols would require a substantial change to the cellular protocols. Another approach is to time-bound the validity of the `system information` messages, by estimating the time required by an adversary to receive a message from a legitimate base-station and then re-transmit it to the UEs [42]. This solution does not take into account differences in the operating frequencies and the coverage area of the base-stations. 5G base-stations can be categorized as wide area, medium area and local area [7], which correspond to macro-cells (500-2500 meters), micro-cell (100-250 metres) and pico-cells (10-50 metres), respectively.

To protect against relay attacks prevention for all those scenarios, we propose time-bounding the `system information` message signatures based on the particular base-station that is broadcasting it. The time-period for which the digital signature generated by a base-station is valid for is denoted by $\triangle \mathsf{t}$. It is calculated using the transmission frequency specific delay $\triangle \mathsf{t}_{\mathsf{freq}}$ and a signature scheme specific cryptographic delay $\triangle \mathsf{t}_{\mathsf{sign}}$. $\triangle \mathsf{t}_{\mathsf{freq}}$ is calculated by the base-station by a lookup table securely stored in the base-station's memory. This lookup table contains the frequency bands and the corresponding $\triangle \mathsf{t}_{\mathsf{freq}}$ values, and is installed and maintained inside the base-stations by their network operators. $\triangle \mathsf{t}_{\mathsf{sign}}$ varies according to the cryptographic scheme used by the base-station to calculate the signature and accounts for the delay caused by the signature scheme. We leave further exploration of these values as future work.

$$\triangle \mathsf{t} = \triangle \mathsf{t}_{\mathsf{freq}} + \triangle \mathsf{t}_{\mathsf{sign}}$$

When signing, the base-station includes the current timestamp $\mathsf{T}_{\mathsf{sign}}$ to denote when the message is being signed and the signature validity time-period $\triangle \mathsf{t}$ with the `system information` message. Upon receiving the message, the UE first checks if the signature is still valid by calculating if $\mathsf{T}_{\mathsf{current}} < \mathsf{T}_{\mathsf{sign}} + \triangle \mathsf{t}$. Here, $\mathsf{T}_{\mathsf{current}}$ is the timestamp at the time of verification.

7 EVALUATION

We not provide the implementation details for the testbed used to evaluate our protocol and report the experimental results on its computational and communication overhead. We also compare our protocol authentication protocols suggested by 3GPP [4].

7.1 Testbed Setup

Cellular network setup. We create a testbed to mimic a generic cellular network with a legitimate base-station, a UE and the 5G core network. The UE is initially connected to a base-station and the core network through it. We then deploy another base-station and move the UE out of the range of the first base-station and close to the second base-station, so that the UE would attempt to connect to this new base-station. Before initiating the connection, the UE listens to the `system information` messages broadcast by the new base-station and uses our authentication protocol to authenticate the SIB1 message and determine whether the new base-station is a legitimate or a fake base-station. The computational and communication overhead results from this testbed should be similar for 5G and 4G LTE networks as the `system information` messages are fairly similar for these generations of cellular protocols.

Hardware and software components. Two popular open-source implementations of the 4G LTE and 5G cellular protocols are OpenAirInterface [11] and srsLTE [15]. However, currently they do not have the implementation for the 5G SIB messages. Since the initial bootstrapping procedures–focus of this work–for both 4G LTE and

5G networks are identical, we leverage srsLTE for our evaluation. The results are worst-case approximations since 5G provides higher bandwidth and has lower communication overhead than 4G LTE.

To build our testbed, we use two *USRP B210* [16] software-defined-radio (SDR) boards connected to two laptops (Intel Core i5 processor at 2.4 GhZ and 8 GB DDR4 RAM) via a USB3 interface. The laptops runs an Ubuntu 18.04 desktop OS. We load srsENB (for the base-station) and srsEPC (for core-network) applications from the srsLTE library [15] on the laptops. The *USRP B210* boards serve as the two base-stations for our testbed. Another *USRP B210* board connected to a similar laptop configuration as the previous ones serves as the UE for our testbed. This laptop runs the srsUE application from the srsLTE library to mimic the working of a UE.

We modify the srsEPC, srsENB and srsUE applications to include our authentication protocol for signing and verifying the SIB1 message. We could not use a commercial mobile device as a UE for our experiments as most of these devices have closed-source firmware which cannot be modified to run our protocol. However, we estimate the performance of our algorithms on a cellular device by measuring the CPU cycles on our setup and converting it to a cellular device clock speed. We provide it in the Appendix Table 3. We use the PBC-library [12] for implementing pairing-based schemes and the FourQlib [10] for FourQ elliptic curves.

7.2 Evaluation Results

Signature Schemes. We consider 5 signature schemes for qualitative and quantitative comparisons with our scheme. Following the recommendations of 3GPP [4], we evaluate identity-based signature schemes BLS [25], BLMQ from IEEE standard 1363.3 [1] and SM9 [27]. We believe that BLS has been mentioned as an identity-based scheme by mistake. While one can devise an identity-based signature from BLS, such schemes are deemed to be very expensive (see Section 3.3 in [45]). 3GPP also recommends the ECCSI scheme from RFC-6507 [38], which is based on the improved schemes proposed by Arazi et al. [19, 20]. However, to the best of our knowledge, there is no provable security argument for these schemes, and the earlier version of such schemes have been shown to be insecure [39]. We thus omit this scheme from our comparisons. We include ECDSA [44] in our comparisons because of its wide adoption. We also include SCRA-BGLS [64], since it is a recent proposal for base-station authentication in 5G [42]. We have implemented the pairing-based schemes with the PBC-library [12] curve d224 which provides 112-bits of symmetric key security. These schemes are even slower for 128 bit security level. All other schemes provide 128-bit symmetric key security according to NIST recommendations [9]. Table 2 summarizes our comparisons.

Quantitative comparison: We evaluate signing and verification costs, end-to-end cryptographic delay, and total communication overhead of the compared schemes. We also implement and compare `Schnorr-HIBS-P`, which uses pre-computed random tokens.

Signing cost: Signature generation in `Schnorr-HIBS-P` with pre-computed tokens only takes 0.009 ms which is 6x faster than SCRA-BGLS, the second fastest scheme. Generating 1000 random tokens takes only 16.35 ms, so these can be generated offline periodically and stored for use during the signing phase. `Schnorr-HIBS-P` also outperforms ECDSA by 52x. All the other schemes have a much slower signing phase. Keeping the signing time low is critical for base-stations as new SIB1 messages are broadcast 160 ms. A low signing overhead should also incentivize the network operators to configure the base-stations to sign all broadcast messages providing full integrity protection.

Verification cost: Our scheme also has the fastest verification phase taking just 0.52 ms. Our scheme outperforms the ECDSA by 2.68x and the SCRA-BGLS scheme by 157x. The verification phase is performed by UEs, which are usually resource constrained devices, so keeping a low verification overhead is critical for saving energy, thus extending the battery life. Moreover, as 5G cellular networks are expected to deploy small base-stations with much smaller coverage areas, UEs would be forced to switch between base-stations at a much faster rate than with conventional base-stations. The presence of (large numbers of) base-stations with small coverage areas would require the UEs to execute the verification phase for the SIB1 message of base-stations at a much higher rate, so keeping the verification overhead low is critical for 5G cellular networks.

End-to-end cryptographic delay: We calculate the total cryptographic overhead of the signature schemes instantiated for cellular base-station authentication, consisting of the signing and verification cost. The end-to-end delay for certificate-based schemes also includes the verification of the sender's public key authenticity via certificates provided by a certification authority. To be favorable to these schemes, we only consider a certificate-chain of size 1. Our scheme provides the lowest end-to-end delay with only 0.53 ms, making it a very good candidate for authentication especially in the presence of base stations with small coverage areas. All the other schemes are much slower than our scheme in terms of the end-to-end cryptographic delay.

Communication overhead: SCRA-BGLS has the smallest signature of 29 bytes. For BLS and SM9, the signature is 48 bytes, whereas our scheme `Schnorr-HIBS-P` and ECDSA-224 have a signature of size 64 bytes. Our scheme and ECDSA also have the smallest public-key size of 32 bytes. Radio wireless channels are a limited resource, making any cryptographic scheme adding a lot of communication overhead to be unlikely to be adopted by network operators. Our scheme provides the smallest communication overhead for authentication. It requires attaching the `Schnorr-HIBS` signature (32 bytes), base-station's ID (9 bytes), base-station's public-key (32 bytes), AMF's ID (7 bytes), AMF's public-key (32 bytes) along with the signing timestamp (4 bytes) and the signature validity period $\triangle t$ (2 bytes) for relay attack prevention. This is a total overhead of 150 bytes, which is much lower than the communication overhead for ECDSA-based PKI (277 bytes) and SCRA-BGLS based PKI (220 bytes). The base-stations can just send the public-keys and the IDs with the SIB1 message which can be cached by the UE. To integrity-protect subsequent messages, our authentication protocol incurs a communication overhead of only 38 bytes per message.

Qualitative comparison: We compare the schemes based on the type of authentication system and the type of scheme.

System: A hierarchical construction is crucial for an identity-based signature scheme suitable for the cellular network architecture (see discussion in Section 6.1). Our scheme is the only identity-based that has a hierarchical construction. Signature based schemes do

Scheme	**Sign**		**Verify**		**Signature**	**PK**	**Crypto E2E**	**System**	**Scheme**
	ms	Cycles	ms	Cycles	(B)	(B)	**Delay** (ms)		**Type**
BLS [25]†	0.62	2.48	6.33	25.97	48	96	13.28	Flat	CB
ECDSA-256 [44]	0.47	1.88	1.40	5.89	64	32	3.28	Flat	CB
SCRA-BGLS [64]	0.06	0.23	82	342.81	29	85	164	Flat	CB
BLMQ [1]	0.62	2.48	6.95	29.07	80	288	7.57	Flat	IDB
SM9 [27]	0.93	3.72	4.10	17.10	48	96	5.03	Flat	IDB
Schnorr-HIBS	0.02	0.05	0.52	2.00	64	32	0.54	Hierarchical	IDB
Schnorr-HIBS-P‡	0.009	0.02	0.52	2.00	64	32	0.53	Hierarchical	IDB

All sizes are in bytes, and all computations are in milliseconds. We also represent the number of CPU cycles for computation in millions. **Signature** and **PK** represent the signature size and public size, respectively. **Scheme Type** indicates whether the scheme is certificate based (CB) or identity-based (IDB). **Crypto E2E Delay** for certificate-based schemes includes the verification of the sender's public key authenticity via certificates provided by a CA. To be favorable to certificate-based schemes, we only consider a certificate-chain of size 1. We implemented the pairing-based schemes with the PBC-library [12] curve d224 which provides 112-bits of security. These schemes are even slower for 128-bit security level. All other schemes provide 128-bit security according to NIST recommendations [9]. † BLS is listed as an identity-based scheme by 3GPP [4] but it is certificate-based (see Section 7.2). ‡ Schnorr-HIBS-P is optimized with pre-computed tokens.

Table 2: Quantitative and qualitative comparison of the candidate signature schemes for authenticating cellular base-stations.

not have a hierarchical construction but support multiple signer levels using certificate-chains.

Type of scheme: BLS, ECDSA and SCRA-BGLS are certificate-based schemes and require a costly public key infrastructure. On the other hand BLMQ, SM9 and `Schnorr-HIBS` are identity-based schemes and are more lightweight than the certificate based solutions as they do not require sending huge certificates.

8 RELATED WORK

To address the attacks where the fake base-stations tamper with the non-integrity protected unicast messages [4, 56, 57] or send reject messages [4, 40, 58], many solutions propose adding integrity protection to these messages or only send these messages after the UE and base-station have established a security context providing confidentiality and integrity protection [4]. However, even after implementing such defenses, the attacker could still carry out these attacks by forcing the UE to use an older vulnerable cellular protocol (4G LTE, 3G, 2G) by using bidding-down attacks.

Other defense techniques rely on improving fake base-station detection and then blacklisting them. One such solution is to use measurement reports sent by UEs to detect inconsistencies between the tampered information being broadcast by the fake base-stations and the legitimate base-station deployment information like the base-station identifier or operation frequencies of the base-stations [6]. Other techniques rely on machine-learning solutions [34, 43, 62] or gathering surrounding network signal statistics from the UEs, legitimate base-stations or other newly deployed hardware [18, 30, 31, 48, 59]. Such techniques can be easily bypassed and have been shown to enable attacks resulting in degradation of network performance and blacklisting of legitimate base-stations [56]. Moreover, many such techniques do not detect the *SigOver* attack [63] that involves overwriting the wireless signals from legitimate base-stations using an attacker with slightly higher signal strength. In addition, these detection techniques cannot prevent the fake base-stations from carrying out the attacks but rather provide a way to detect the fake base-stations after-the-fact. Most of those defense techniques patch the specific vulnerabilities of individual protocols used by the fake-base stations, but do not tackle the root cause behind them.

3GPP has proposed three possible solutions allowing UEs to authenticate base-stations [4]: ❶ Verification of SI messages using digital signatures or identity-based cryptography using keys provisioned by the network operators. This solution, however, is applicable only for the verification of the authenticity of the base-station during `RRC_IDLE` mode and `RRC_INACTIVE` mode cell re-selection. The UE is not able to authenticate the base-station during the initial registration procedure. ❷ Use Certification Authority (CA) based PKI to assign certificates to base-stations for signing their broadcast messages. However, sending the CA certificates along with the broadcast messages has a large communication overhead and the ECDSA scheme proposed for this solution adds a lot of computational overhead at the UE for verifying the certificate chain as well as the signature itself. ❸ Using an identity-based signature scheme to sign the base-station's broadcast messages. These schemes are more lightweight than the certificate based solutions as they do not require the base-station to send large certificates; however, the suggested identity-based schemes BLS [25], BLMQ from IEEE standard 1363.3 [1] and SM9 [27] require pairing computations on the verifiers' side, considered a highly expensive cryptographic operation. ECCSI signature scheme from RFC-6507 [38] lacks a security proof and earlier versions of it have been shown to be insecure [39].

Other PKI-based solutions proposed for the authentication of base-stations have induce a high communication and computational overhead on the cellular network and the UE [47, 65, 66]. An optimised PKI-based authentication solution has been recently proposed; however, this scheme has a high verification overhead at the UE [42]. This solution works for 4G LTE networks where the base-stations usually cover large areas and the UEs rarely have to switch the base-stations they are connected to. However, due to the smaller coverage area of the 5G base-stations, the UEs will have to switch the base-stations they are connected to much more frequently. As a consequence, the UEs have to verify the `system information` messages very frequently resulting in a high cumulative verification overhead that can cause latency issues.

9 DISCUSSION

Relay attacks. Our time-bounding technique for relay attack protection is a best-effort approach. It cannot completely thwart relay attacks, rather it raises the bar for the attackers. A precise defense for relay attack would require a sophisticated approach with major changes in the protocol (e.g., including new messages) and a precise estimation of the timing/latency of message transmission and calculation, environmental interference, and hardware used by base stations and cellular devices. We leave this for future work.

Lawful interceptions. 5G enables a separate interface at the core network [2] for lawful interceptions which remains unchanged by our scheme. Law enforcement agencies can also request temporary public keys for base-stations from the cellular service providers for lawful interceptions. Our scheme prevents illegitimate interceptions exploiting the lack of authentication of SIB broadcast messages.

Emergency services. According to 3GPP [6], devices without SIM/USIM/eSIM do not perform authentication with the network for emergency calls/SMSs. To authenticate base-stations in cases where a SIM was installed in the cellular device but was subsequently removed, the device can use a cached copy the public key of their network operator's Core-PKG in the UICC. We do not support base-station authentication if a SIM was never inserted in a cellular device.

UICC vs. UE. 3GPP recommends either UICC or UE for public-key encryption of SUPI [6]. We, therefore, envision the implementation of Schnorr-signature verification at UE.

Backward compatibility. Our solution is backward compatible since base-stations only include the information required for our scheme in the *optional fields* of SIB messages. Legacy devices ignore those fields and authentication.

10 CONCLUSION AND FUTURE WORK

We have proposed an efficient authentication protocol for 5G networks based on a hierarchical identity-based signature scheme. Our protocol achieves at least 6 times speedup in terms of end to end cryptographic delay over the 3GPP proposals for authenticating base-stations. Our protocol also achieves a communication cost reduction of 31% over the other schemes. As future work, we will minimize the communication and computation overheads incurred by our authentication protocol when providing integrity protection to all messages transmitted by base-stations. We will also develop a robust relay attack prevention mechanism by better estimating the transmission delay by leveraging machine learning techniques.

ACKNOWLEDGEMENT

Part of this work is supported by NSF Award #1917627.

REFERENCES

[1] 2013. IEEE Approved Draft Standard for Identity-Based Cryptographic Techniques using Pairings. *IEEE P1363.3/D8, April 2013* (2013), 1–136.

[2] 2018. 3GPP, Specification number TS 33.127 version 15.0.0, Release Description; Lawful Interception (LI) architecture and functions.

[3] 2019. 3GPP, Specification number TR 21.915 version 15.0.0, Release Description; Release 15.

[4] 2020. 3GPP, Specification number TR 33.809 version 0.8.0, Study on 5G security enhancements against false base stations.

[5] 2020. 3GPP, Specification number TS 29.571 version 15.6.0, Common Data Types for Service Based Interfaces.

[6] 2020. 3GPP, Specification number TS 33.501 version 15.8.0, Security architecture and procedures for 5G System.

[7] 2020. 3GPP, Specification number TS 38.104 version 15.9.0, Base Station (BS) radio transmission and reception.

[8] 2020. 3GPP, Specification number TS 38.331 version 15.9.0, Radio Resource Control (RRC); Protocol specification.

[9] 2020. BlueKrypt: Cryptographic Key Recommendation. https://www.keylength.com/en/4/.

[10] 2020. FourQlib. https://www.microsoft.com/en-us/research/project/fourqlib/.

[11] 2020. OpenAirInterface. https://www.openairinterface.org/?page_id=2761.

[12] 2020. PBC: Pairing-Based Cryptography Library. https://crypto.stanford.edu/pbc/.

[13] 2020. Ping Time Between Los Angeles and New York. https://wondernetwork.com/pings/Los%20Angeles/New%20York.

[14] 2020. ProVerif: Cryptographic protocol verifier in the formal model. https://prosecco.gforge.inria.fr/personal/bblanche/proverif/.

[15] 2020. srsLTE. https://github.com/srsLTE/srsLTE.

[16] 2020. USRP B210. https://www.ettus.com/all-products/UB210-KIT/.

[17] Martín Abadi and Cédric Fournet. 2001. Mobile values, new names, and secure communication. *ACM Sigplan Notices* 36, 3 (2001), 104–115.

[18] Hamad Alrashede and Riaz Ahmed Shaikh. 2019. IMSI Catcher Detection Method for Cellular Networks. In *2019 2nd International Conference on Computer Applications & Information Security (ICCAIS)*. IEEE, 1–6.

[19] Ortal Araz and Hairong Qi. 2006. Load-balanced key establishment methodologies in wireless sensor networks. *International Journal of Security and Networks* 1, 3-4 (2006), 158–166.

[20] Ortal Arazi, Itamar Elhanany, Derek Rose, Hairong Qi, and Benjamin Arazi. 2006. Self-certified public key generation on the intel mote 2 sensor network platform. In *2006 2nd IEEE Workshop on Wireless Mesh Networks*. 118–120.

[21] Mihir Bellare and Gregory Neven. 2006. 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, CCS 2006, Alexandria, VA, USA, Ioctober 30 - November 3, 2006*. 390–399.

[22] Mihir Bellare and Philip Rogaway. 1993. Random oracles are practical: A paradigm for designing efficient protocols. In *Proceedings of the 1st ACM conference on Computer and Communications Security (CCS '93)* (Fairfax, Virginia, United States). ACM, NY, USA, 62–73.

[23] Daniel J. Bernstein, Peter Birkner, Marc Joye, Tanja Lange, and Christiane Peters. 2008. Twisted Edwards Curves. In *Progress in Cryptology - AFRICACRYPT 2008, First International Conference on Cryptology in Africa, Casablanca, Morocco, June 11-14, 2008. Proceedings*. 389–405.

[24] Alexandra Boldyreva, Adriana Palacio, and Bogdan Warinschi. 2012. Secure Proxy Signature Schemes for Delegation of Signing Rights. *J. Cryptology* 25, 1 (2012), 57–115.

[25] Dan Boneh, Ben Lynn, and Hovav Shacham. 2001. Short signatures from the Weil pairing. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 514–532.

[26] Sanjit Chatterjee, Chethan Kamath, and Vikas Kumar. 2012. Galindo-Garcia Identity-Based Signature Revisited. In *Information Security and Cryptology - ICISC 2012 - 15th International Conference, Seoul, Korea, November 28-30, 2012, Revised Selected Papers*. 456–471.

[27] Zhaohui Cheng. [n.d.]. The SM9 Cryptographic Schemes. ([n. d.]).

[28] Sherman S. M. Chow, Lucas Chi Kwong Hui, Siu-Ming Yiu, and K. P. Chow. 2004. Secure Hierarchical Identity Based Signature and Its Application. In *Information and Communications Security, 6th International Conference, ICICS 2004, Malaga, Spain, October 27-29, 2004, Proceedings*. 480–494.

[29] Craig Costello and Patrick Longa. 2015. Four$\mathbb{Q}$: Four-Dimensional Decompositions on a $\mathbb{Q}$-curve over the Mersenne Prime. In *Advances in Cryptology – ASIACRYPT 2015*, Tetsu Iwata and Jung Hee Cheon (Eds.). Springer Berlin Heidelberg, 214–235.

[30] Adrian Dabrowski, Georg Petzl, and Edgar R Weippl. 2016. The messenger shoots back: Network operator based IMSI catcher detection. In *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer, 279–302.

[31] Adrian Dabrowski, Nicola Pianta, Thomas Klepp, Martin Mulazzani, and Edgar Weippl. 2014. IMSI-catch me if you can: IMSI-catcher-catchers. In *Proceedings of the 30th annual computer security applications Conference*. 246–255.

[32] Danny Dolev and Andrew Yao. 1983. On the security of public key protocols. *IEEE Transactions on information theory* 29, 2 (1983), 198–208.

[33] Ulrich Dürholz, Marc Fischlin, Michael Kasper, and Cristina Onete. 2011. A formal approach to distance-bounding RFID protocols. In *International Conference on Information Security*. Springer, 47–62.

[34] Paal Engelstad, Boning Feng, Thanh van Do, et al. 2016. Strengthening mobile network security using machine learning. In *International Conference on Mobile Web and Information Systems*. Springer, 173–183.

[35] Dan Forsberg, Günther Horn, Wolf-Dietrich Moeller, and Valtteri Niemi. 2012. *LTE security*. John Wiley & Sons.

[36] David Galindo and Flavio D. Garcia. 2009. 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*. 135–148.

[37] Craig Gentry and Alice Silverberg. 2002. Hierarchical ID-Based Cryptography. In *Advances in Cryptology - ASIACRYPT 2002, 8th International Conference on the*

Theory and Application of Cryptology and Information Security, Queenstown, New Zealand, December 1-5, 2002, Proceedings. 548–566.

[38] Michael Groves. 2012. *Elliptic Curve-Based Certificateless Signatures for Identity-Based Encryption (ECCSI)*. RFC 6507.

[39] Isabelle Hang, Markus Ullmann, and Christian Wieschebrink. 2011. Short Paper: A New Identity-Based DH Key-Agreement Protocol for Wireless Sensor Networks Based on the Arazi-Qi Scheme. In *Proceedings of the Fourth ACM Conference on Wireless Network Security* (Hamburg, Germany) *(WiSec '11)*. Association for Computing Machinery, New York, NY, USA, 139–144. https://doi.org/10.1145/1998412.1998436

[40] Syed Hussain, Omar Chowdhury, Shagufta Mehnaz, and Elisa Bertino. 2018. LTEInspector: A systematic approach for adversarial testing of 4G LTE. In *Network and Distributed Systems Security (NDSS) Symposium 2018*.

[41] Syed Rafiul Hussain, Mitziu Echeverria, Omar Chowdhury, Ninghui Li, and Elisa Bertino. 2019. Privacy Attacks to the 4G and 5G Cellular Paging Protocols Using Side Channel Information.. In *NDSS*.

[42] Syed Rafiul Hussain, Mitziu Echeverria, Ankush Singla, Omar Chowdhury, and Elisa Bertino. 2019. 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*. 1–11.

[43] Jian Jin, ChangLiang Lian, and Ming Xu. 2019. Rogue Base Station Detection Using A Machine Learning Approach. In *2019 28th Wireless and Optical Communications Conference (WOCC)*. IEEE, 1–5.

[44] Don Johnson, Alfred Menezes, and Scott Vanstone. 2001. The elliptic curve digital signature algorithm (ECDSA). *International journal of information security* 1, 1 (2001), 36–63.

[45] Eike Kiltz and Gregory Neven. 2009. Identity-Based Signatures. In *Identity-Based Cryptography*. 31–44. https://doi.org/10.3233/978-1-58603-947-9-31

[46] Denis Foo Kune, John Koelndorfer, Nicholas Hopper, and Yongdae Kim. 2012. Location leaks on the GSM air interface. *ISOC NDSS (Feb 2012)* (2012).

[47] Cheng-Chi Lee, I-En Liao, and Min-Shiang Hwang. 2009. An extended certificate-based authentication and security protocol for mobile networks. *Information Technology and Control* 38, 1 (2009).

[48] Zhenhua Li, Weiwei Wang, Christo Wilson, Jian Chen, Chen Qian, Taeho Jung, Lan Zhang, Kebin Liu, Xiangyang Li, and Yunhao Liu. 2017. FBS-Radar: Uncovering Fake Base Stations at Scale in the Wild.

[49] Andy Lilly. 2017. IMSI catchers: hacking mobile communications. *Network Security* 2017, 2 (2017), 5–7.

[50] Huang Lin. 2016. LTE REDIRECTION: Forcing Targeted LTE Cellphone into Unsafe Network. In *Hack In The Box Security Conference (HITBSec-Conf)*.

[51] Stig F Mjølsnes and Ruxandra F Olimid. 2017. Easy 4G/LTE IMSI catchers for non-programmers. In *International Conference on Mathematical Methods, Models, and Architectures for Computer Network Security*. Springer, 235–246.

[52] Chris Paget. 2010. Practical cellphone spying. *Def Con* 18 (2010).

[53] Kasper Bonne Rasmussen and Srdjan Capkun. 2010. Realization of RF Distance Bounding.. In *USENIX Security Symposium*. 389–402.

[54] David Rupprecht, Katharina Kohls, Thorsten Holz, and Christina Pöpper. 2019. Breaking LTE on layer two. In *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1121–1136.

[55] Claus Schnorr. 1991. Efficient Signature Generation by Smart Cards. *Journal of Cryptology* 4, 3 (1991), 161–174.

[56] Altaf Shaik, Ravishankar Borgaonkar, Shinjo Park, and Jean-Pierre Seifert. 2018. On the impact of rogue base stations in 4g/lte self organizing networks. In *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. 75–86.

[57] Altaf Shaik, Ravishankar Borgaonkar, Shinjo Park, and Jean-Pierre Seifert. 2019. New vulnerabilities in 4G and 5G cellular access network protocols: exposing device capabilities. In *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks*. 221–231.

[58] Altaf Shaik, Jean-Pierre Seifert, Ravishankar Borgaonkar, N. Asokan, and Valtteri Niemi. 2016. Practical Attacks Against Privacy and Availability in 4G/LTE Mobile Communication Systems. In *23rd Annual Network and Distributed System Security Symposium, NDSS 2016, San Diego, California, USA, February 21-24, 2016*. The Internet Society. http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2017/09/practical-attacks-against-privacy-availability-4g-lte-mobile-communication-systems.pdf

[59] Simen Steig, Andre Aarnes, Thanh Van Do, and Hai Thanh Nguyen. 2016. A network based imsi catcher detection. In *2016 6th International Conference on IT Convergence and Security (ICITCS)*. IEEE, 1–6.

[60] Daehyun Strobel. 2007. IMSI catcher. *Chair for Communication Security, Ruhr-Universität Bochum* 14 (2007).

[61] Nils Ole Tippenhauer, Heinrich Luecken, Marc Kuhn, and Srdjan Capkun. 2015. UWB rapid-bit-exchange system for distance bounding. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. 1–12.

[62] Thanh Van Do, Hai Thanh Nguyen, Nikolov Momchil, et al. 2015. Detecting IMSI-catcher using soft computing. In *International Conference on Soft Computing in Data Science*. Springer, 129–140.

[63] Hojoon Yang, Sangwook Bae, Mincheol Son, Hongil Kim, Song Min Kim, and Yongdae Kim. 2019. Hiding in Plain Signal: Physical Signal Overshadowing Attack on {LTE}. In *28th USENIX Security Symposium (USENIX Security 19)*. 55–72.

[64] Attila Altay Yavuz, Anand Mudgerikar, Ankush Singla, Ioannis Papapanagiotou, and Elisa Bertino. 2017. Real-time digital signatures for time-critical networks. *IEEE Transactions on Information Forensics and Security* 12, 11 (2017), 2627–2639.

[65] Xun Yi, Eiji Okamoto, and Kwok Yan Lam. 1998. An optimized protocol for mobile network authentication and security. *ACM SIGMOBILE Mobile Computing and Communications Review* 2, 3 (1998), 37–39.

[66] Yuliang Zheng. 1996. An authentication and security protocol for mobile computing. In *Mobile Communications*. Springer, 249–257.

A PROOF OF THEOREM 5.1

PROOF. $\mathcal{R}_1$ captures the case where $\mathcal{A}$ makes at least one signature query during the simulation phase for the target identity and the embedded challenge value is included in the output of O_{Sign}. In this reduction algorithm, the ECDLP instance aP is embedded in the commitment value of the signature R instead of the key. $\mathcal{R}_1$ uses the generalized forking lemma [21] and the knowledge of the private key of all the users, except the target user ID^*, to obtain a set of two congruences and two unknowns to solve for a. This is done by partitioning the identity space into two disjoint sets $\mathcal{I}_{Ex}$ and $\mathcal{I}_S$ randomly. The reduction algorithm can respond to both extract and signature queries for the identities in $\mathcal{I}_{Ex}$. However, for identities in $\mathcal{I}_S$, it aborts on any extract queries, but it can respond to signature queries.

The reduction algorithm $\mathcal{R}_2$ uses the multiple forking lemma [24] and captures the complement of the above event. More precisely, $\mathcal{R}_2$ captures the case where $\mathcal{A}$ does not make a signature query on the target user ID^*, or Q_{ID^*} was never returned as a part of the signature query for ID^*. Therefore, the ECDLP challenge is embedded in the master public-key (*mpk*). For this event, we need to assume that $\mathcal{A}$ makes a query on H_2 for ID^*, before a H_1 query is made on ID^*. To respond to O_{Sign} queries, C uses it accesses to the random oracles to simulate the signatures (similar to the simulation of Schnorr signatures) on ID^*. However, note that if final forgeries of $\mathcal{A}$ contain Q_{ID^*} from such sign algorithm, it will not contain the answer to the ECDLP. However, the assumption above $\mathcal{R}_2$, ensures this does not happen. This reduction uses multiple-forking lemma [24] to solve for a.

$\mathcal{R}_3$ algorithm works in the event that $\mathcal{A}$ does not make a signature query on ID^*, or Q_{ID^*} was never returned as a part of the signature query for ID^* and $\mathcal{A}$ makes a query on H_1 for ID^*, before a H_2 query is made on ID^*. The main difference between $\mathcal{R}_2$ and $\mathcal{R}_3$ is that in $\mathcal{R}_3$, the multiple-forking lemma is invoked for four iterations to obtain a set of four congruences and four unknowns to solve for a.

Note that the simulation of the reduction algorithms above will be indistinguishable from the real world, by utilizing the random oracles to respond to $\mathcal{A}$'s queries.

B FORMAL VERIFICATION

We verify the correctness of our authentication protocol using an automated cryptographic protocol verifier, ProVerif [14]. We consider the Dolev-Yao model [32] (see Section 4). ProVerif takes as input the description of our proposed protocol specifications in applied π-calculus dialect [17], and the security and privacy properties, such as secrecy, authenticity and observational equivalences that we want to verify. ProVerif specifications in applied π-calculus are then translated into corresponding Horn clauses and property verification is performed through logical derivation by applying resolution techniques used in logic programming.

Modeling choices. We model the legitimate participants, such as cellular device, base-station, AMF, and the core-PKG in the core network as processes. We use a term algebra to model messages, and equational theory to model algebraic properties of cryptographic primitives. We consider an infinite set of names to represent keys, IDs, and nonces.

Properties. We evaluate the following security properties: - Secrecy for the private-keys of the core-PKG, AMF and base-station.

- Weak authentication (i.e., correspondence) and strong authentication (i.e., injective-correspondence) properties to verify the authenticity and replay protections for the SIB1 message broadcast by base-station to the cellular device.
- Authenticity of the base-station's key to the cellular device to ensure that the key is generated by the legitimate AMF.
- Authenticity of the AMF's key to the cellular device to ensure that the key is generated by the legitimate core-PKG.

Results. We provide the code for the Proverif formal verification below. ProVerif indeed provided no counter-examples for the above properties and thus signifies the correctness of our proposed authentication protocol.

B.1 Proverif Code

```
(* Communication channels between PKG, AMF, BS and UE*)
free pkg_to_amf: channel [private].
free amf_to_bs: channel [private].
free bs_to_ue: channel.

(*=============================================*)
(*=============================================*)

(*Data Types*)
type nonce.
type public_key.
type secret_key.
type ID.

(*=============================================*)
(*=============================================*)

(*Functions*)

(* Get public key for a particular secret key *)
fun get_public_key(secret_key): public_key.

(* Generate public key from a nonce *)
fun generate_public_key(nonce): public_key.

(* Generate secret key based on ID, parent secret key and nonce *)
(* This function is abstracted for simplicity*)
fun generate_secret_key(secret_key, ID, nonce): secret_key.

(* Sign a message using Schnorr-HIBS scheme *)
fun hibs_sign(bitstring, secret_key): bitstring.


(*=============================================*)
(*=============================================*)

(* Destructors *)

(* Verify the signature of a message*)
reduc forall m: bitstring, k: secret_key;
        checksign(hibs_sign(m, k), get_public_key(k)) = m.

(* Verify the public key using parent public key*)
reduc forall k: secret_key, xID: ID, rand: nonce;
        parentkey(get_public_key(generate_secret_key(k,
                                    xID, rand))) = k.


(*=============================================*)
(*=============================================*)

(* Secrecy queries *)

(*Attacker should not have access to the private keys*)
free secret_key_pkg, secret_key_amf,
        secret_key_bs: secret_key [private].

query attacker(secret_key_pkg);
   attacker(secret_key_amf);
   attacker(secret_key_bs).


(*=============================================*)
(*=============================================*)

(* Authentication queries *)
event authentication_successful(public_key).
event begin_signing(public_key).


query x: public_key; inj-event(authentication_successful(x))
        ⟹ inj-event(begin_signing(x)).


(*=============================================*)
(*Core_PKG process*)
let Core_PKG(secret_key_pkg: secret_key) =
        (*Generate Keys for AMF*)
        in(pkg_to_amf, ID_AMF: ID);
        new rand: nonce;
        let public_key_amf = generate_public_key(rand) in
                out(pkg_to_amf, (public_key_amf,
                generate_secret_key(secret_key_pkg,
                ID_AMF, rand))).

(*=============================================*)
(*AMF process*)
```

```
let AMF() =
        (*Get new keys generated by Core_PKG*)
        new ID_AMF: ID;
        out(pkg_to_amf, ID_AMF);
        in(pkg_to_amf, (public_key_amf: public_key,
                        secret_key_amf: secret_key));

        (*Generate keys for the Base-stations*)
        in(amf_to_bs, ID_BS: ID);
        new rand: nonce;
        let public_key_bs = generate_public_key(rand) in
            out(amf_to_bs, (public_key_bs,
            generate_secret_key(secret_key_amf, ID_BS, rand),
                ID_AMF, public_key_amf )).

(*=============================================*)

(*Base-station process*)
let BS() =
        (*Get new keys generated by AMF*)
        new ID_BS: ID;
        out(amf_to_bs, ID_BS);
        in(amf_to_bs, (public_key_bs: public_key,
                secret_key_bs: secret_key, ID_AMF: ID,
                public_key_amf: public_key));

        (*Sign the SIB1 message*)
        new SIB1: bitstring;
        event begin_signing(public_key_bs);
        let sig_SIB1 = hibs_sign(SIB1, secret_key_bs)  in
                out(bs_to_ue, (SIB1, sig_SIB1, ID_BS, ID_AMF,
                        public_key_bs, public_key_amf)).

(*=============================================*)

(*UE process*)
let UE(public_key_pkg: public_key) =
        (*Receive and verify signature*)
        in(bs_to_ue, (SIB1: bitstring, sig_SIB1:bitstring,
            ID_BS: ID, ID_AMF: ID, public_key_bs: public_key,
            public_key_amf: public_key));

        (* Check whether signature is valid*)
        let (= SIB1) = checksign(sig_SIB1, public_key_bs) in
        (* Verify BS's public-key*)
        let(= public_key_amf) = get_public_key(
                        parentkey(public_key_bs)) in
        (* Verify AMF's public-key*)
        let(= public_key_pkg) = get_public_key(
                parentkey(public_key_amf)) in
        event authentication_successful(public_key_bs).

(*=============================================*)
(*=============================================*)
(* Main process*)
process
        (* Generate PKG's secret key*)
        new secret_key_pkg: secret_key;

        (* Start all individual processes in parallel*)
        (Core_PKG(secret_key_pkg) |
                !AMF() |
                !BS() |
                !UE(get_public_key(secret_key_pkg)))
```

C EVALUATION ON A CELLULAR DEVICE

Scheme	Sign		Verify	
	ms	Cycles	ms	Cycles
BLS [25]	1.24	2.48	12.98	25.97
ECDSA-256 [44]	0.94	1.88	2.94	5.89
SCRA-BGLS [64]	0.11	0.23	17.14	342.81
BLMQ [1]	1.24	2.48	14.53	29.07
SM9 [27]	1.86	3.72	8.55	17.10
Schnorr-HIBS	0.02	0.05	1	2.00
Schnorr-HIBS-P	0.01	0.02	1	2.00

We estimate the signing and verification performance of all the schemes by assuming a cellular device with a CPU clock speed of 2 GHz. All computations are in milliseconds and CPU cycles are in millions.

Table 3: Comparison of the signing and verification of the candidate signature schemes for authenticating cellular base-stations on cellular device.