Insecure Connection Bootstrapping in Cellular Networks: The Root of All Evil

Syed Rafiul Hussain
Purdue University
hussain1@purdue.edu

Mitziu Echeverria
University of Iowa
mitziu-echeverria@uiowa.edu

Ankush Singla
Purdue University
asingla@purdue.edu

Omar Chowdhury
University of Iowa
omar-chowdhury@uiowa.edu

Elisa Bertino
Purdue University
bertino@purdue.edu

ABSTRACT

In the cellular ecosystem, base stations act as trusted intermediaries between cellular devices and the core network. During connection bootstrapping, devices currently, however, do not possess any mechanisms to authenticate a base station before connecting to it. This lack of authentication has been shown to be exploitable by adversaries to install fake base stations which can lure unsuspecting devices to connect to them and then launch sophisticated attacks. Despite being a well-known threat to the cellular ecosystem, this weakness is not addressed in the current protocol versions including 5G. The current paper sets out to fill this void by proposing a Public-key infrastructure (PKI) based authentication mechanism which builds on top of the asymmetric cryptography used in 5G and adheres to the relevant deployment constraints. Our proposed authentication scheme leverages precomputation-based digital signature generation algorithms and employs optimizations in three dimensions—PKI scheme-level, protocol-level, and cryptographic scheme-level—to address the trilemma of small signature size, efficient signature generation, and short verification time. Our evaluation on a real testbed indicates that the proposed scheme is not only readily deployable but also performs better than a symmetric key-based scheme (i.e., TESLA) in terms of security guarantee, overhead, and deployment constraints (e.g., backward compatibility).

CCS CONCEPTS

• **Security and privacy → Mobile and wireless security**.

KEYWORDS

Cellular Network, 4G LTE, 5G, Broadcast, Authentication

ACM Reference Format:
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 *12th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '19), May 15–17, 2019, Miami, FL, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3317549.3323402

WiSec '19, May 15–17, 2019, Miami, FL, USA

ACM ISBN 978-1-4503-6726-4/19/05…$15.00
https://doi.org/10.1145/3317549.3323402

1 INTRODUCTION

A cellular device's connection to the operator's network starts off by the device scanning for network signals broadcast by nearby base stations. Among the available base stations, the device selects to initiate the connection to the one that emits signals with the highest strength. Once the device connects to the base station, the base station then plays the role of a trusted intermediary enabling the device to seamlessly communicate with the core network.

Unfortunately, no mechanism currently exists by which a device can verify the legitimacy of a base station [41]. This lack of authentication allows adversaries to install rogue base stations which lure unsuspecting devices to connect to them [34, 44]. Forcing devices to connect to a fake base station is often the necessary first step for the adversary to carry out other destructive attacks, such as man-in-the-middle [10, 11, 20, 30], location tracking [12, 23, 44], SMS phishing [31], relay [22, 42], and denial-of-service [22, 44] attacks. Although this fundamental connection bootstrapping weakness is widely acknowledged, there does not seem to be a conscious effort in mitigating this even in the 5G standard [1] which only strives to protect illegitimate exposures of a device's permanent identifier using public-key encryption. *This paper aims to fill this gap by proposing an authentication mechanism that can be retrospectively added to the existing protocol for securing the connection bootstrapping process between cellular devices and base stations.*

Existing work: Unlike a majority of the existing research [12, 13, 21, 22, 37, 39, 42, 45, 46], which focuses on identifying security weaknesses of cellular networks, there are only a few proposals that focus on misbehaving base stations [19, 29, 33, 52]. The most relevant is the proposal by Li et al. [29], named FBS-Radar, which collects spam messages (and, accompanying meta-data) received by end-users to identify locations of fake base stations. In the same vein, Zhuang et al. [52] developed FBSleuth which uses Radio Frequency Fingerprinting to establish forensic evidence of a base station's misbehavior. These prior efforts, however, cannot alleviate the root cause (i.e., *insecure bootstrapping*) that allows adversaries to lure devices into connecting to fake base stations.

Challenges: A clean-slate bootstrapping authentication mechanism that requires major protocol/infrastructural overhaul or breaks backward compatibility is unlikely to be embraced by existing stakeholders due to its associated deployment cost. In the context of deployability, thus a mechanism that can be retrospectively incorporated into existing deployments without violating deployment constraints and backward compatibility is required. More

concretely, any effective mechanism has to decide for which bootstrapping signals of the base station it will provide authentication guarantees while taking into consideration the quality of service, overhead of base stations and cellular devices, bandwidth, scheduling constraints, and maximum transmission unit (MTU).

Approach: Conceptually, one can consider the following two high-level approaches for bootstrapping authentication: ones based on the *TESLA* [36] protocol and the others based on *Public Key Infrastructure* (PKI). TESLA is a broadcast authentication protocol in which multiple time-synchronized receivers (i.e., cellular devices) can authenticate messages periodically broadcast by a sender (i.e., base station). TESLA uses symmetric cryptographic functions (MAC) to achieve asymmetric properties using delayed key disclosure and hash chains. However, due to the delayed key disclosure, the receivers have to buffer the messages and wait for the corresponding signing key to be released by the sender before the verification process can be completed, which adds a significant overhead. TESLA also requires an authenticated channel for its own bootstrapping, making it ineffective without an enabling PKI.

In the PKI-based approach, which we prescribe in this paper, each base station is equipped with a public, private key pair. Any broadcast signal emitted by a specific base station will be digitally signed by its private key. Any cellular device that has access to the base station's public key can then verify that the bootstrapping signal is indeed emitted by the claimed base station. To realize such an approach, the cellular device is required to first verify the authenticity of a base station's public key. This can be achieved by the base station sending a public-key certificate chain (e.g., X.509 [17]).

At face value, implementing the PKI-based authentication mechanism seems like a straightforward proposition. In reality, however, realizing an implementation of the authentication mechanism requires addressing several deployment constraints. Due to the restriction on the broadcast packet size, we observed that even a single vanilla X.509 certificate does not fit into a packet. Another relevant issue one needs to consider for a successful deployment is the revocation of a base station's public key prior to its expiration date, especially, in exceptional cases (e.g., private key leak). This is a relevant threat as the adversary can gain physical access to the base stations which are often left unguarded. Typical revocation mechanisms (e.g., CRL [17] and OCSP [43]) are ineffective in our context as they require connectivity which the device is attempting to gain in the first place. The final challenge one would have to address is to protect against relay or replay attacks which have been shown to be extremely effective in case of cellular networks [22, 42]. We address these challenges in the following manner.

Certificate size: To overcome the packet size constraint, we use a custom encoding of a certificate containing only fields that are relevant to our context (e.g., identity, public-key, expiration).

Revocation: We avoid an explicit revocation mechanism by proposing base station certificates to have a small but configurable expiration time (e.g., <10 minutes) that limits the attack window.

Relay-/replay-attack protection: We introduce a location-dependent, configurable parameter that influences the validity period of a given broadcast message and in turn can control the exploitation window.

The final issue we need to address for the realization of an effective and secure authentication mechanism is the choice of a digital signature scheme. Such a choice impacts three different aspects: (a) Signature size; (b) Signature generation time; (c) Signature verification time. This is also known as the trilemma of digital signatures and a scheme can only minimize two of these aspects. We choose to optimize aspects (a) and (b) while sacrificing (c). The rationale of optimizing aspect (a) is clear as this will minimize the overhead of the packet size. We optimize aspect (b) instead of aspect (c) because a device typically will verify signatures for only a small number of sessions whereas the base station keeps generating signatures for the bootstrapping signals based on its schedule (e.g., ~80 milliseconds). Optimizing aspect (b) will decrease the computation and energy overhead for the base station.

Implementation: We implemented our PKI-based broadcast authentication for 4G LTE (since no open-source 5G implementation is available) in a real test-bed using software-defined radios and open-source 4G LTE protocol stack [7, 9]. For digital signature schemes, we consider ECDSA [24], BGLS [15], and SCRA-BGLS [48] (BGLS signature generation optimized with offline pre-computation). In our evaluation, we observed that our mechanism imposes only moderate overhead with respect to additionally transmitted bytes (e.g., ~**220** bytes) and connection time (e.g., ~**120** milliseconds).

Impact: In the recent 5G proposal, public-key cryptography is already introduced for protecting against illegitimate exposure of IMSI through IMSI-Catching attacks [3, 34, 44] by requiring devices to encrypt their IMSIs/IMEIs with the network operator's public key stored in the device. Using this root of trust, our solution builds a PKI on top of it and can be seen as an add–on to the existing 5G public-key cryptography, enabling devices to authenticate base stations and prevent many different attacks. As 5G is still awaiting deployment, incorporating a defense such as ours is feasible, and can go a long way in securing the cellular ecosystem.

Contributions: In summary, the paper has the following contributions.

- We propose an optimized PKI-based authentication mechanism that enables a cellular device to authenticate a base station during connection bootstrapping. Our defense can protect against many high-profile attacks against the cellular network including the notorious IMSI-catching attack and DNS redirection through man-in-the-middle relays.
- We implemented our scheme on a real test-bed using software-defined radios and open-source protocol stack.
- Our evaluation on a real test-bed shows that our approach incurs a moderate overhead with respect to the number of transmitted bytes, signature generation time for the base station, and connection establishment overhead for the device.

2 BACKGROUND

In this section, we present an overview of the cellular network architecture, and cell selection and initial bootstrapping procedures.

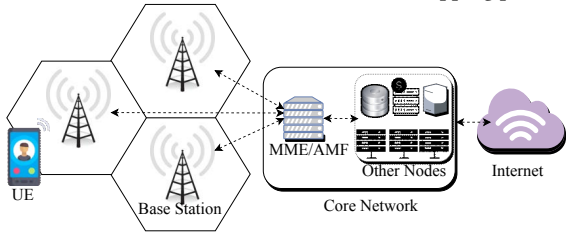

Figure 1: Cellular network architecture

2.1 Cellular Network Architecture

Cellular networks (as shown in Figure 1) can be broken down into three different components: cellular device, also known as User Equipment **(UE)**. Radio Access Network, referred to as **(RAN)** and the core network.

UE: It is the cellular device equipped with a universal subscriber identity module known as SIM card. The SIM card securely stores the unique international mobile subscriber identity known as IMSI in 4G (resp. SUPI in 5G) and its associated cryptographic keys used for the UE identification and authentication during the UE's connection initiation with the core network. The UE also has its own device-specific unique identity, called international mobile equipment identity (IMEI) also used for identification. The IMSI and IMEI are sensitive in the sense that exposing them can make the UE prone to tracking/impersonation.

RAN: A geographical area, in the context of a cellular network, is partitioned into hexagonal cells (see Figure 1) where each cell is serviced by a single base station, providing its nearby cellular devices connectivity to Internet through the carrier's core network. In essence, the RAN is the network between a UE and a base station, and between pairs of base stations.

EPC: The core network consists of many components among which the Mobility Management Entity referred to as **MME** (resp., Access and Mobility Management Function (AMF) in 5G) is the most relevant for our work. The MME/AMF manages the critical sub-protocols, such as attach (UE's initial connection to the core network), paging (used for notifying UE's pending services), and detach (disconnection from the core network). The MME/AMF is also in charge of keeping track of locations of UEs and is interchangeably used throughout in the paper.

2.2 UE's Cell Selection and Bootstrapping

A base station periodically broadcasts frame synchronization signals/messages (from the physical layer), and `master_info_block` (**MIB**) at 40 milliseconds interval and `system_info_block` (**SIB**) messages (from the upper layer) at 80 milliseconds interval to advertise the existence of the network irrespective of any user's presence in a cell area. The cellular device scans the frame synchronization signals broadcast by nearby base stations in the frequency bands that the device is allowed to operate on and for each frequency it identifies the strongest among all the suitable/acceptable cells. A suitable/acceptable cell is the one for which the measured cell attributes satisfy the cell selection criteria. When an acceptable cell is found, the UE camps on that cell and initiates the cell reselection procedure, if required. The UE reads the MIB message sent by the selected cell, and synchronizes the time. The UE learns the connection-related parameters' values from the SIB messages after which it initiates connection (as shown in Figure 2) to the base station (at the radio resource control or RRC layer) and to the core network (at Network Access Stratum or NAS layer).

3 PROBLEM DESCRIPTION

In this section, we first present our adversary model, then formulate the problem we address in this paper.

3.1 Threat Model

In our threat model, the adversary has the following capabilities:

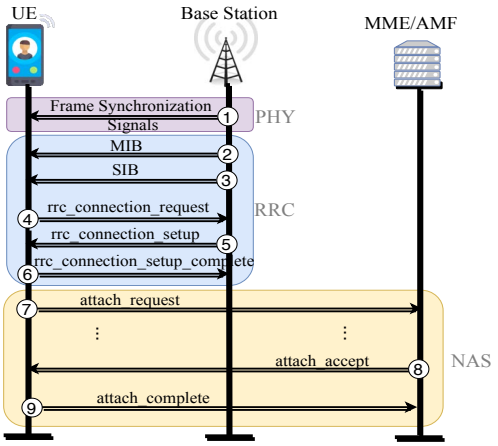

Figure 2: Cell selection and initial connection setup with the base station and the MME/AMF of the core network.

Eavesdropping or tampering with protocol messages. We consider the adversary to have the capability of establishing a man-in-the-middle relay [22, 42] which in turn may allow him to drop, modify, eavesdrop, and forward messages transmitted between benign protocol participants (e.g., legitimate devices and base stations) in the public channel while respecting cryptographic assumptions.

Impersonating a legitimate base station. We also consider an active adversary who can install and run its own base station with the same capabilities as a legitimate one. In addition, the fake base station can impersonate a legitimate base station and thus can force a victim device to connect to it by broadcasting MIB and SIB messages in the victim UE's frequency with a higher signal strength than the legitimate base station. We make the assumption that the adversary can learn legitimate values for MIB and SIB messages by eavesdropping the public channels where these are broadcast.

Other assumptions. We assume that the adversary cannot physically tamper with the SIM card, base station, or the core network to obtain the sensitive information, e.g., cryptographic master or session keys. Side-channel attacks and denial-of-service (DoS) attacks due to wireless signal jamming are considered out of scope.

3.2 Scope and Problem Statement

Lack of authentication of MIB and SIB messages enables the adversary to spoof a legitimate base station. The adversary exploiting this deeply rooted vulnerability can lure an unsuspecting cellular device to connect to it and then carry out specific attacks using unauthenticated messages exclusively sent to the victim device. We identify the following two types of defense mechanisms that could prevent such attacks. (i) *Attack-specific defenses* attempt to thwart a particular discovered vulnerability. For instance, ignoring unauthenticated and out-of-order `auth_reject` messages can protect devices from denial-of-service attacks as demonstrated by Hussain et al. [22]. (ii) *Generic defenses*, on the contrary, prevent the root cause of a vulnerability which may be exploited by multiple attacks. In our context, such a defense would be to prevent adversaries from forcing the UE to connect to the fake base station in the first place by making possible for the UE to authenticate base stations. Naturally, the former types of defenses protect devices only from a very specific set of *discovered* attacks due to adversary's use of fake base stations. There are, however, many other attacks that exploit the capability of setting up a fake base station [22]; such

an attack-specific defense cannot thwart these and hence such a case-by-case defense cannot be a practical solution. It is thus clear that because of its wider applicability, a generic defense mechanism is critical for the security of cellular networks. Designing such a mechanism is the focus of the paper.

Protocol versions. Our discussion, although mainly focusing on the 4G LTE and 5G versions of the cellular protocol, is generalizable to older protocol versions (e.g., 3G/2G).

Which messages to authenticate? A cellular device may authenticate either (i) the broadcast MIB and SIB messages, or (ii) the exclusive connection setup message (i.e., the `rrc_connection_setup` message in step 5 of Figure 2) to connect to a legitimate base station. When the device is in the idle mode (i.e., no radio activity), however, it only captures the MIB and SIB messages and camps on a cell for receiving paging messages without setting up any explicit connection with the base station. Since paging messages along with MIB and SIB messages do not have any integrity/authenticity protection, the adversary can inject fake emergency alerts using fabricated SIB and paging messages [22]. Such attacks in device's idle mode are hard to prevent without ensuring the authenticity of MIB and SIB messages. Authenticating the `rrc_connection_setup` message, on the other hand, implicitly requires authenticating the SIB messages (through MAC) received before. Since at bootstrapping time the UE and the base station do not share a session key, it is not clear which key to use for generating the MAC. In light of the above discussion, it is therefore clear that broadcast (i.e., MIB, SIBs) messages are the natural choice for authenticating a base station and the root of trust for establishing a secure connection to the base station and consequently with the core network.

Problem Statement. Formally, in this paper, we aim to design and evaluate a secure connection bootstrapping mechanism for cellular devices by providing authentication guarantee and integrity protection to MIB and SIB broadcast-messages while conforming to the constraints of cellular ecosystems.

4 POTENTIAL SOLUTIONS

In this section, we discuss two candidate mechanisms that can possibly provide the authenticity and integrity protection for bootstrapping signals/messages transmitted by the base stations. One of these mechanisms is based on symmetric key cryptography whereas the other is based on public key cryptography. Conceptually, both mechanisms are capable of providing cellular devices the necessary method for authenticating base stations. We discuss their relative merits and demerits in the context of deployment.

4.1 Infeasible Symmetric Key-based Mechanisms

One infeasible but straightforward symmetric key based approach for providing broadcast authentication is to use Message Authentication Codes (MAC) [25]. At its core, a MAC-based authentication mechanism can provide the integrity protection for the broadcast messages without incurring substantial computational or space overhead. Having an effective MAC-based authentication mechanism, however, boils down to effective key management. Sharing a single symmetric key between all devices and base stations is not viable as the adversary can extract the key and subsequently bypass the security. Having a pre-shared key between each pair of device and base station, on the other hand, is infeasible with respect to key management and storage requirements.

Another promising symmetric key based authentication mechanism is the TESLA protocol [36] which addresses the broadcast authentication problem by achieving asymmetric key properties. Due to its promised security guarantees, we qualitatively analyze it with respect to the deployment constraints of cellular networks.

4.1.1 TESLA Protocol Description. TESLA solves the broadcast authentication problem in the scenario where a single sender (i.e., base station) delivers a message to multiple receivers (i.e., UEs) using a pre-determined time schedule. The TESLA protocol can be broken down into four phases: sender setup, receiver bootstrapping, broadcasting of messages, and authentication of messages. In the sender setup phase, the sender constructs a one-way hash chain, divides time into equal intervals, and assigns each interval a key from the one-way hash chain in reverse order of generation. During this phase, the sender also sets the disclosure delay (i.e., how many intervals must pass before a given key is disclosed). In the receiver bootstrapping phase, the receiver must be loosely time-synchronized with the sender and receive an authenticated key from the generated hash chain along with the disclosure delay. Note that for this protocol to be secure the aforementioned information must be transmitted through an *authenticated channel*, typically achieved using digital signatures. Before broadcasting a message, the sender appends two additional items to the packet—the MAC for that message and the key to be disclosed at that time interval. The actual key used to compute the MAC is derived from the key assigned to the time interval after the delay. To verify the messages, the receiver must first buffer the messages that it receives along with its MAC. After the corresponding key has been revealed, the receiver derives the key and verifies the corresponding MAC. To verify the key itself, the receiver verifies if it is part of the hash-chain by comparing it with the authenticated key from the bootstrapping phase.

4.1.2 TESLA for Cellular Networks. For adapting TESLA to our context, we first have to accomplish the required time synchronization between the base station and cellular devices. Fortunately, time synchronization is already provided by cellular networks through the use of MIB messages. The next challenge is to satisfy its assumption of a pre-existing authenticated channel to share the *disclosure delay* and *initial key commitment* with the receivers. This suggests that the base station should include this information in a MIB message while ensuring its authenticity and integrity. Without such guarantees, a fake base station can simply broadcast its own TESLA protocol parameters, which a cellular device will not be able to distinguish from a legitimate one. One possible approach of addressing this challenge is to use some form of digital signatures. This would require the base station to have a public-key whose authenticity can be verified by the device with the use of a PKI and a trust anchor stored in the device. The use of digital signatures and PKI, however, undermines the actual purpose of using a symmetric key based approach. Moreover, due to the disclosure delay, the base station must also send an additional message (e.g., SIB3) or force the UE to wait for the subsequent set of SIB messages disclosing the previous key to complete the verification of previous messages. The former optimizes with respect to delay; however, it requires

additional bytes to be broadcast which does not respect the constraints imposed by the cellular stakeholders. Though the latter does not require an additional message, it unfortunately induces a significant delay (i.e., 80 ms). In summary, due to its reliance on an *authenticated channel*, latency and communication overhead, we conclude that TESLA is an infeasible mechanism for cellular ecosystems.

4.2 PKI-based Mechanism

In this section, we briefly present a high-level overview of a PKI-based secure bootstrapping mechanism that we envision and then outline the challenges one has to address for achieving an optimal PKI-based broadcast authentication scheme for cellular networks.

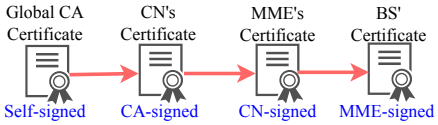

Figure 3: Initial PKI Scheme.

4.2.1 High-level Overview of PKI-based Solution. In our initial attempt at developing a PKI-based solution, we consider the following stakeholders: the core network/network operator (denoted with CN); the MME; the base station (denoted with BS). We consider each of these entities to possess a public-key, secret-key pair of the form $\langle P, S \rangle$. Each of these entities also has a public-key certificate (e.g., X.509 certificate) which maps its identity to the public-key (see Figure 3). The BS' certificate is digitally signed by the secret key of the MME (i.e., S_{MME}). Similarly, the MME's certificate is digitally signed by the CN's private key (i.e., S_{CN}). We also consider a global certificate authority (CA) which also has a self-signed certificate which will be stored in the device's memory and will be used as the trust anchor, that is, the CA's secret key will be used to digitally sign the CN's certificate. Once such a PKI is established, it is possible to provide a mechanism through which a device can authenticate a base station.

In such a mechanism (see Figure 4), for authenticating a specific bootstrapping message m, the base station (i.e., BS_j) using its secret key S_{BS_j} will generate the signature sig_m of m and will append the signature (e.g., sig_{SIB} for a SIB message) and the certificate-chain $\langle sig_m, cert_{BS_j}, cert_{MME_i}, cert_{CN} \rangle$ to m. Once the device receives m, its digital signature, and the certificate chain, it will first verify the certificate chain and then will verify m's digital signature. Legacy devices in which signature verification mechanisms are not present or will be too demanding, on the other hand, can safely ignore both the signature and certificate chain.

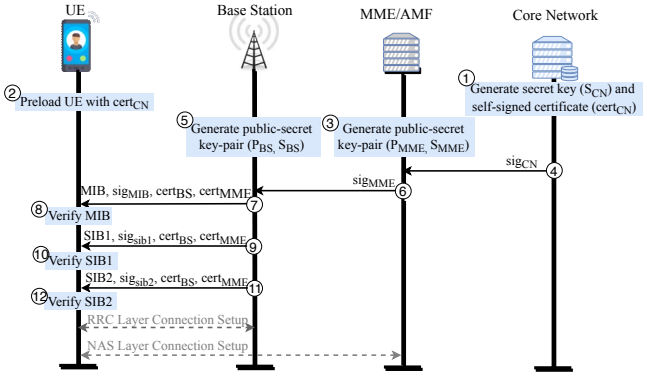

Figure 4: Unoptimized PKI Scheme.

Deployment challenges. Realizing the above straightforward mechanism in practice, however, requires us to address the following challenges. (i) There is an upper-limit on the size of MIB and SIB messages which imposes an upper-limit on the size of the certificate chain. Since the size of a X.509-based certificate [17] is prohibitively large, it is nearly impossible to fit the X.509 certificate-chain in a single MIB/SIB message. We have empirically validated this claim. (ii) It is not clear how would one facilitate certificate revocation in our setting. (iii) The broadcast signals along with the digital signature can be relayed/replayed by a man-in-the-middle (MitM) relay attacker possibly luring devices to connect to a fake base station. (iv) The base station frequently broadcasts the MIB and SIB messages (e.g., SIB is sent every 80 ms) and hence to maintain this transmission schedule the packet construction overhead including the signature generation time should be minimized.

5 OPTIMIZED PKI SCHEME

In this section, we discuss how we address the above challenges by optimizing the proposed scheme from three dimensions: (1) PKI-level, (2) protocol-level, and (3) cryptographic scheme-level.

5.1 PKI-level Optimizations

Realigning trust anchor. As the device inherently has to trust the core network, one can provision the SIM card to use the core network's certificate as the trust anchor instead of the global CA's certificate. This has the added benefit of decreasing the certificate chain length which in turn reduces the message size and computation time for verifying the chain. This, however, leads to the problem where a device attempts to authenticate base stations not signed by its core network (i.e., roaming). To alleviate this scenario, the SIM card can be equipped with (or, delivered over-the-air) the certificates belonging to the roaming network operators. This is particularly feasible due to the recent introduction of eSIM cards [4] which can be overwritten with software by the provider. We envision that an eSIM is provisioned with the root certificate of the core network (i.e., the home network operator). In the instance of roaming, the user may communicate with the core network through an off-band channel to update the root certificate.

A Lightweight Design of Certificate. A general X.509 certificate is equipped with many different fields and extensions which are not relevant to our context and hence can be omitted. We propose a specialized certificate format only containing the following fields:

$$cert_{CN} = P_{CN}, MCC, MNC, ext_{cert_{CN}}$$
$$cert_{MME_i} = P_{MME_i}, MME\_ID, ext_{cert_{MME_i}}, sig_{CN}$$
$$cert_{BS_j} = P_{BS_j}, CELL\_ID, loc_{BS_j}, ext_{cert_{BS_j}}, sig_{MME_i}$$

where, MCC and MNC form the unique network ID, MME_ID and CELL_ID respectively represent the unique identities of MME_i and BS_j, loc_{BS_j} denotes the physical location (i.e., latitude and longitude) of BS_j, and $ext_{cert_{CN}}$ and $ext_{cert_{MME_i}}$ indicate the certificate expiration time for CN and MME_i, respectively. The core network's signature for MME_i (i.e., sig_{CN}), and the MME_i's signature for BS_j (i.e., sig_{MME_i}) are computed as follows:

$$sig_{CN} = sign(\langle P_{MME_i}, MME\_ID, ext_{cert_{MME_i}} \rangle, S_{CN})$$
$$sig_{MME_i} = sign(\langle P_{BS_j}, CELL\_ID, loc_{BS_j}, ext_{cert_{BS_j}} \rangle, S_{MME_i})$$

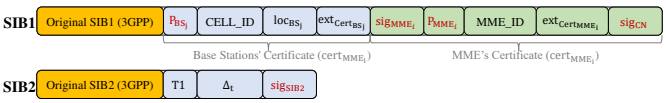

Figure 5: Content of SIB1 and SIB2 after protocol-level optimization for secure broadcast authentication

where the $sign(m, K)$ function generates the digital signature of a given message m with respect to a secret key K.

Certificate revocation. Instead of an explicit revocation mechanism (e.g., CRL, OCSP), we rely on short-lived certificates (e.g., <10 minutes) for the base stations. The validity period of the certificates is left as an implementation parameter and its value influences the exploitation window size when base stations are compromised.

5.2 Protocol-level Optimizations

Since including authentication material increases both the packet size and computational time on the base station and device sides, we minimize the overhead by: (a) Providing authentication guarantees for only critical messages; (b) Aggregating authentication of multiple messages; (c) Limiting the certificate chain transmission.

5.2.1 Authenticating Critical Broadcast Signals. Ideally, all broadcast messages should be authenticated; however, such an approach can be impractical due to its substantial communication and computational overhead requirement. We thus only provide authentication guarantees for a limited number of bootstrapping messages.

Messages requiring authentication. During bootstrapping, frame synchronization signals and MIB provide instructions on how to decode subsequent SIB messages and thus enable devices to achieve current system frame time synchronization with the base station. The SIB messages, on the other hand, provide information necessary for connection establishment. Therefore, if only SIB messages are authenticated instead of frame synchronization and MIB messages, an adversary may only launch a DoS attack by jamming the network by desynchronizing the frames, which, as mentioned earlier, is out of scope. Thus, we do not provide authentication for frame synchronization and MIB messages and induce minimal changes to SIB messages only. Note that other signaling messages are not subject to any changes with our proposed scheme.

Which SIBs to authenticate?. According to the 3GPP standard, there are 13 System Information Block (SIB) messages (i.e., SIB1-13) characterized by the type of information they carry. SIB1 provides the essential information regarding the radio access network (RAN) and includes a broadcast schedule for subsequent SIB messages. Authenticating SIB1 thus guarantees that devices will obtain legitimate access information along with a legitimate broadcast schedule. Otherwise, the adversary could inject fake scheduling information as well as fake SIBs which would enable him to broadcast fake emergency alerts using SIB1 and SIB10-11 messages. Therefore, SIB1 requires authentication and integrity protection.

SIB2 immediately follows SIB1 and provides the necessary information for initiating the attach procedure (i.e., initial connection with the base station and the core network). Since SIB2 contains critical information for connecting to the base station, we provide authentication for this message. Protection for the other SIBs is not mandatory as they are not critical to connection bootstrapping.

5.2.2 Minimizing certificate chain transmission. Since SIB1 and SIB2 are sent in the same radio frame, the probability of one of the certificates in the chain getting revoked between SIB1 and SIB2 is negligible. The base station thus transmits the certificate chain with the SIB1 message only, and expects the device to use the same base station public key for authenticating the SIB2 signature. In the extreme case, that is, when revocation happens between SIB1 and SIB2, the base station can include the new certificate chain in SIB2 and its presence can be indicated by a single bit.

Figure 6: Content of SIB1 and SIB2 after cryptographic scheme-level optimization for secure broadcast authentication

5.2.3 Aggregating authentication. Since a base station broadcasts SIB1 and SIB2 in the same radio frame (1 radio frame = 10 ms) but in a different subframe (1 subframe = 1 ms), and the cellular device does not initiate a connection with the base station before receiving SIB2, we propose to authenticate SIB1 and SIB2 messages together instead of individually. Thus, the base station includes the certificate-chain in the SIB1 message whereas the digital signature authenticating SIB1 and SIB2 (i.e., sig_{SIB2}) is sent with the SIB2 message (as shown in Figure 5). Precisely, $sig_{SIB2} = sign(\langle SIB1 \parallel SIB2 \rangle, S_{BS_j})$.

The device, therefore, should buffer the SIB1 message and verify both messages only after the reception of the SIB2 message. The device is also required to verify the certificate-chain included in the SIB1 message using $cert_{CN}$ provisioned in the SIM card. This aggregated authentication of SIB1 and SIB2 reduces the time, computational resources, and communication overhead otherwise incurred by individual authentications of SIB1 and SIB2 messages.

5.3 Cryptographic scheme-level Optimization

The choice of digital signature schemes can not only influence the security provided by a mechanism but also the overhead incurred due to the size of the signature, time to generate and verify a signature. As mentioned before, we aim to maintain respectable security (i.e., 112-bit) while optimizing for signature sizes and signature generation time. In what follows, we discuss possible signatures schemes and their effectiveness in our context.

RSA and ECDSA. RSA [40] is one of the most widely used signature schemes in the wild. In our context, RSA, however, is inappropriate as it requires a large key and generates a large signature when maintaining our desired 112-bit of security. In our evaluation, we observed that RSA keys and signatures are too large to fit in either SIB1 or SIB2 messages. Elliptic Curve Digital Signature Algorithm (ECDSA) [14] scheme, on the other hand, is a viable replacement of RSA as it can provide the same level of security with a smaller key and signature size compared to RSA. ECDSA signature generation and verification, however, incur a significant computational overhead due to its inherent expensive cryptographic operations.

BGLS. We also considered BGLS [16] which has two desired properties: (1) It generates fairly small signatures while maintaining the desired level of security; (2) It allows the aggregation of multiple digital signatures—generated from different private keys—into a single short signature. Property (2) of BGLS especially comes in handy for aggregating the signatures in the certificate chain (see Figure 6) which consequently reduces the communication overhead (in bytes). BGLS, however, incurs a substantial overhead at the verifier side due to expensive cryptographic pairing operations.

SCRA-BGLS: To further reduce the computation overhead of BGLS signature generation, we leverage the *Structure-free and Compact*

Real-time Authentication (SCRA) framework [49] which divides the message signing operation into offline and online stages. It shifts the expensive parts of the signature generation algorithm to the offline key-generation phase. The online signature generation phase leverages the pre-computed values from the offline phase and performs lightweight cryptographic operations.

We now briefly discuss how the SCRA algorithm works for generating SIB2's signature (i.e., sig^{agg}_{SIB2}).

1) Key Generation (Offline): The offline stage is executed just once when the base station starts. A d-bit hash of SIB1 || SIB2 can be thought of as L equal chunks of b bits each such that $b \cdot L = d$. A signature is computed for each b-bit integer concatenated with its corresponding index i and a predefined padding using the base station's secret-key S_{BS_j}. A pre-computed sub-message/signature table Γ is generated and stored at the sender's side.

2) Signature Generation: For the SIB2 message, the sender computes the cryptographic hash of SIB1 || SIB2 and divides it into L chunks. It then fetches the corresponding signatures from the table Γ. Finally, it combines the signatures from the pre-computed table efficiently according to the base scheme.

3) Signature Verification: Upon reception of SIB1 and SIB2, the cellular device computes the hash of SIB1 || SIB2 and runs the verification algorithm on the signature and the hash using the public keys P_{BS_j}, P_{MME_i}, and P_{CN} among which P_{BS_j} and P_{MME_i} are included in SIB1 message and P_{CN} is provisioned in the SIM card.

5.4 Countermeasure for Relay Attacks

Since not all control-plane cellular protocol messages are cryptographically protected, a fake base station relaying/replaying bootstrapping messages from a legitimate base station can lure devices to connect to it and then launch different attacks [22, 42]. Digital signatures alone cannot protect against such threats. For thwarting such attacks, one would ideally need to deploy a distance-bounding protocol [38] which, however, would require substantial change to the protocol. Thus, we adopt a best effort approach by allowing each bootstrapping message to be valid for only a short period of time limiting the attack opportunity and raising the bar for attackers.

In our approach, we consider each SIB2 message to contain the following three additional fields: T_{gen} denoting the time at which the message was constructed; a location-dependent parameter Δ_t; loc_{BS_j} denoting the latitude and longitude of the base station. If a device receives an SIB2 message at time T_i, it would consider it valid if and only if $T_i - T_{gen} < \Delta_t$. A fake base station can only successfully relay a legitimate SIB2 if it can ensure that the relayed message reaches the device within Δ_t time of T_{gen}. Under an appropriate Δ_t value, due to triangle inequality, one can minimize the threat of a relay attack (see Figure 7).

Computing the exact value for Δ_t is non-trivial as it requires taking location-dependent signal interference into consideration which is hard to approximate due to environmental dynamics. We, however, show how to approximate lower and upper bounds of Δ_t which we envision a base station would calculate periodically. Our bound calculation requires the following constants.

- C_{BS} : The time difference between when an SIB2 is generated (T_{gen}) and transmitted. A significant portion of this time will likely be spent on signature generation.
- C_S: Transmission time of SIB2 from memory to network.
- C_R: Time required by a device to receive and store a SIB2 message.
- C_{ADV}: Time required by an adversary to alter the contents of a legitimate message.
- R : Base station's broadcast radius in some unit d.
- S : Time required to travel one unit of distance (d = 1) at the speed of light.

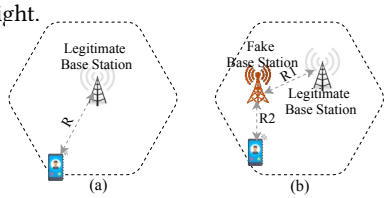

Figure 7: Illustration for calculating the (a) lower and (b) upper bound of Δ_t.

Lower bound: The lower bound of Δ_t (denoted by Δ_{TL}) can be approximated by the maximum time required by an SIB2 message to travel from a legitimate base station to a UE. This would require the UE to be located at the farthest point from the legitimate base station within the cell. Figure 7a represents the scenario described above. In such a case, the network delay can be computed as $(R * S)$; note that, S is not the speed of light. Hence, $\Delta_{TL} = C_{BS} + (R * S) + C_S + C_R$.

Upper bound: The upper bound of Δ_t (denoted by Δ_{TU}) can be approximated by the minimum time required by a fake base station to successfully relay/replay a legitimate SIB2 message to a device. This scenario is shown in Figure 7b. In this case, suppose that the distance between the legitimate and fake base station is R_1 whereas the distance between the fake base station and the device is R_2. As the SIB2 message is sent twice (once by the legitimate base station and then by the fake base station requiring a time of $2C_S$) and received twice (once by the fake base station and then by the cellular device), the device requires $2C_R$ to store and receive a relayed SIB2 message. Also, the adversary must modify the contents of the bootstrapping message which requires a time of C_{ADV}. Therefore, the total time required for the SIB2 message to travel from a legitimate base station to the device through the fake base station can be computed as: $\Delta_{TU} = C_{BS} + ((R_1 + R_2) * S) + 2(C_S + C_R) + C_{ADV}$.

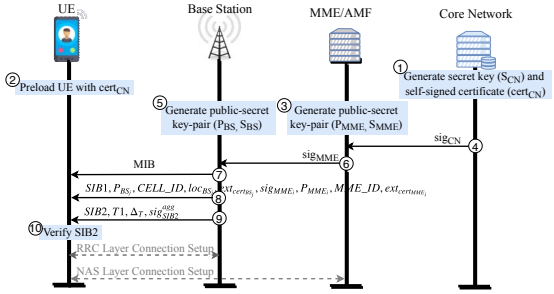

Figure 8: Optimized PKI Scheme.

Selecting a value for Δ_t: As the time required to travel one unit distance, S is inversely proportional to the speed of light (i.e., $S = \frac{1}{3*10^8}$) and is significantly smaller than the values of R and $R1 + R2$, $R * S$ and $(R1 + R2) * S$ can be canceled out from the computation of Δ_{TL} and Δ_{TU}, respectively. Since the fake base station requires an extra round of message transmission and reception along with the time needed to alter the contents of the bootstrapping message, we argue that $\Delta_{TL} << \Delta_{TU}$. Therefore, we need to select a value for Δ_t such that the following condition is satisfied: $\Delta_{TL} < \Delta_t < \Delta_{TU}$. Any relay protection mechanism requires a precise time

synchronization between UEs and a base station. In our scheme, the maximum allowable amount of time drift is C_{ADV}.

Figure 8 summarizes the our proposed optimized PKI scheme.

6 EVALUATION

In this section, we empirically evaluate the effectiveness and incurred overhead of the different instantiations of our scheme.

6.1 Testbed Setup

Setup with 4G LTE (Why not 5G?). We chose to set up the testbed for 4G LTE mainly due to the following reasons: (1) there are currently no open-source implementations of 5G UE, base station, and core network available; and (2) the bootstrapping broadcast signals (i.e., the frame synchronization, MIB, and SIB signals) and the initial connection setup procedures for both 4G and 5G are identical. Hence, the overhead and security guarantees induced by our optimized PKI scheme in 4G LTE will likely transfer to 5G.

Base station and core network setup. We use a USRP B210 [9]) as the hardware component connected to an Intel Core i7 machine running Ubuntu 16.04. We used srsLTE [7], an open-source LTE protocol stack implementation, for establishing the base station and core network. We set up the base station and the core network in the same machine with srsENB [7] and srsEPC, respectively.

UE setup. We use a similar USRP B210 [9] connected to an Intel Core i7 machine running srsUE [8] (open-source UE protocol stack implementation) as the next generation UE which costs around $1300. We enhanced srsUE to follow our mechanism. Note that the computation time on a core i7 machine may not be comparable to an actual cellular device (e.g., feature phones or resource-constrained cellular IoT device), the overhead results reported in this section can be considered as the lower bound overhead. The computation power of machines powering the base station and core network, however, is superior to the i7 machines used in the evaluation.

Why not actual UE? Since commercial modems' firmware are closed source, incorporating our proposed solution is unachievable.

SIB1			
Field	**ECDSA-224**	**BGLS**	**SCRA-BGLS**
MME Public Key	57	85	85
MME Public Key Expiration	4	4	4
MMEI	3	3	3
Base Station Public Key	57	85	85
Base Station Public Key Expiration	4	4	4
MME Signature	64	N/A	N/A
CN Signature	64	N/A	N/A
Total	253	181	181

Table 1: Overhead in bytes per field in the SIB1 message due to extra bytes added for authentication. N/A denotes that the field is not broadcast in SIB1 when using the given scheme.

SIB2			
Field	**ECDSA-224**	**BGLS**	**SCRA-BGLS**
Timestamp	4	4	4
Delta	2	2	2
Longitude	2	2	2
Latitude	2	2	2
SIB1+SIB2 Signature	64	N/A	N/A
Aggregated Signatures	N/A	29	29
Total	74	39	39

Table 2: Overhead in bytes per field in the SIB2 message due to extra bytes added for authentication. N/A denotes that the field is not broadcast in SIB2 when using the given scheme.

6.2 Evaluation Results

We evaluate the effectiveness of our proposed defense with respect to the following metrics. We consider three digital signature schemes: (i) ECDSA-224, (ii) BGLS, and (iii) SCRA-BGLS for comparing our proposed PKI scheme with the baseline implementation which does not include broadcast authentication.

(I) Overhead in bytes. Table 1 and Table 2 show the byte overhead per field in the SIB1 and SIB2 messages for different schemes. Table 1 shows that ECDSA-224 requires SIB1 to include two 64-byte signatures of CN and MME, whereas the BGLS and SCRA-BGLS calls for no signature at all in SIB1 message. Since BGLS and SCRA-BGLS schemes aggregate CN's signature and MME's signature with SIB2's signature, the SIB1 message does not incur additional overhead.

Table 2 shows that the SIB2 message with ECDSA-224 scheme includes one 64-byte signature, whereas both BGLS and SCRA-BGLS add only a 29-byte signature to SIB2. This is due to BGLS' capability of not only generating small signatures but also aggregating multiple signatures. Hence, BGLS and SCRA-BGLS incur substantially lower communication overhead than ECDSA-224.

(II) Signature generation time. Two of the three signatures (i.e., CN Signature, MME Signature) in our optimized PKI scheme are computed by CN and MME offline, and are shared with the base station at base station boot up time. These signatures can be used until the keys expire. Table 3 shows the computation overhead, i.e., the time required for generating CN's and MME's signature offline with four different signature schemes and the corresponding time required for generating base station's signature at runtime. Since SCRA-BGLS takes the lowest signature generation time (0.084 ms) and results in the smallest signature size (29-bytes) among four different schemes, we adopt SCRA-BGLS over other schemes.

Algorithm	**CN Signature**		**MME Signature**		**Base Station Signature**	
	Avg. (ms)	SD (ms)	Avg. (ms)	SD (ms)	Avg. (ms)	SD (ms)
ECDSA-224	1.20	0.01	1.19	0.02	1.21	0.02
BGLS	1.74	0.49	1.92	0.54	3.08	1.08
SCRA-BGLS	0.084	0.007	0.082	0.004	0.084	0.006

Table 3: The average (denoted with Avg.) time taken by the CN, MME, and base station to generate required signatures. CN's and MME's signatures are generated offline whereas base station's signature is generated at runtime. SD refers to standard deviation.

Due to the relay/replay protection, the timestamp and signature of the base station for authenticating SIB1 and SIB2 have to be computed prior to every broadcast. Table 3 demonstrates that SCRA-BGLS incurs the smallest overhead (0.084 ms) compared to the latencies of ECDSA-224 (1.21 ms) and BGLS (3.08 ms) because SCRA-BGLS minimizes the number of expensive crypto operations at runtime by offloading them to the offline phase. Since the base station using BGLS aggregates the three signatures into one signature in SIB2 (without precomputation), the time required to generate the aggregated signature using BGLS is higher than the rest.

(III) Signature verification time. With ECDSA-224, the UE verifies the signatures of CN, MME, and base station individually. Table 4 shows the latency for verifying each ECDSA-224 signature.

Algorithm	**Verify CN's Sig.**		**Verify MME's Sig.**		**Verify base station's Sig.**	
	Avg. (ms)	SD (ms)	Avg. (ms)	SD (ms)	Avg. (ms)	SD (ms)
ECDSA-224	2.27	0.19	2.26	0.21	2.27	0.23

Table 4: The time taken by a UE to verify each ECDSA-224 signature. Avg. stands for average and SD is the standard deviation.

Table 5 shows the total time taken by the UE to verify signatures when using different schemes. The verification time at the UE, however, is significantly higher than the signature generation time (Table 3) at the base station as the base station can reuse precomputed signatures whereas the UE must verify all three signatures separately. To avoid this, when using ECDSA-224, the UE could maintain pairing of signatures and the public keys of base station and MME in memory so that once these are verified, the UE can look them up on a table and avoid signature verification for subsequent messages.

Since, in both BGLS and SCRA-BGLS, the UE has to perform expensive bilinear pairing checks for verifying a signature, they both have significantly higher verification time (17.81 ms and 119.19 ms, respectively) than ECDSA-224 (6.81 ms). Note that in BGLS there are only two pairing calculations whereas in SCRA-BGLS there are 32 pairing calculations; hence SCRA-BGLS incurs the highest verification time among the four schemes. Since the UE typically verifies signatures for one session whereas the base station keeps generating signatures for the bootstrapping signals based on its schedule (e.g., ~80 milliseconds), we have chosen to minimize the overhead of the packet size and signature generation time at base stations. Considering all these trade-offs, we argue that SCRA-BGLS is the most effective digital signature scheme in the context of cellular ecosystem due to both its low packet overhead and low signature generation time.

Algorithm	Verify	
	Average (ms)	Standard Deviation (ms)
ECDSA-224	6.81	0.33
BGLS	17.81	6.0
SCRA-BGLS	119.19	0.9

Table 5: The total time taken by a UE to verify the signatures in the SIB1 and SIB2 messages when using different signature schemes.

(IV) SCRA-BGLS pre-computation overhead. With SCRA-BGLS, the time required to pre-compute the signature table is 5729.36 ± 8.2 seconds and the space required to store that table is 160 KB when the total number of chunks is 32 and each chunk is 8-bits long.

(VI) Energy consumption overhead. As modern UEs are mostly black-boxes, there is no clear way to implement our signature verification scheme on a real UE and calculate the energy overhead. We, therefore, approximate the energy overhead with CPU cycles used by our proposed techniques and compare them to CPU cycles used by the baseline approach. The percentage overhead of our scheme in terms of CPU cycles is a loose indicator of the energy overhead (assuming everything else remains the same, like sensor usage, screen usage, etc.). Our evaluation shows that ECDSA-224, BGLS, and SCRA-BGLS schemes take only 8292, 12484, and 131453 extra CPU cycles (of a core i7 machine), respectively, which induce minimal effect on the battery consumption on modern smartphones [6] as they operate at $2.8 \text{x} 10^8$ cycles/second.

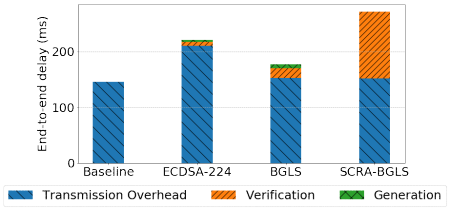

Figure 9: End-to-end delay induced by different digital signature schemes against baseline.

(V) End-to-end delay. We define the end-to-end delay (computed at the UE side) as the time between when an SIB1 message starts being generated and when the UE verifies SIB2 and is ready to set up the RRC layer connection with the base station. Figure 9 compares the baseline and four digital signature schemes with respect to the end-to-end delay. Each stacked bar in Figure 9 shows the transmission overhead (in times), signature generation, and verification times with three different individual segments. Due to large public key and signature, ECDSA-224 induces the highest transmission overhead (~210 ms) which naturally boils down our choice to BGLS and SCRA-BGLS that add negligible transmission overhead compared to the baseline. BGLS-SCRA induces the highest end-to-end delay (~270 ms), and BGLS has the lowest end-to-end delay (~176 ms). Even though SCRA-BGLS has the highest end-to-end delay, based on the trilemma we argue that it is the most optimized PKI scheme in the context of cellular network.

(VI) TESLA. We also evaluate the TESLA-based bootstrapping scheme adapted for cellular networks and compare it with our scheme. For TESLA, we generate a one-way hash chain of 10,000 keys which takes only ~14 ms. To complete the receiver bootstrapping, we used what we deemed as the most optimized PKI scheme (SCRA-BGLS) which leads to an overhead of 220 bytes and ~270 ms end-to-end delay. To verify the message the UE takes 0.0028 ms. To obtain the key, we use the subsequent SIB messages; this is done to reduce the overhead in bytes. It, however, induces a significant increase in verification time (~80 ms). An additional overhead of 64 bytes is required with each message to accommodate the MAC (32 bytes) and the disclosed key (32 bytes). In total, TESLA adds a delay of ~350 ms and a ~284 byte overhead. In summary, TESLA is not a feasible option in our scenario as our best scheme performs as well as the setup phase while avoiding the additional overhead.

7 SECURITY ANALYSIS

We now analyze the security guarantees of our PKI-based countermeasure with respect to our adversary model (Section 3.1).

- *Injection/modification of SIB messages.* Since the adversary does not know the legitimate base station's private key, it will not be able to generate a valid signature of a fake SIB message protecting UEs from fake base stations. Similarly, the authentication will fail with our solution in place if an on-path attacker using MitM relay modifies the contents of a legitimate SIB1 message. Since the UE eventually verifies SIB1 along SIB2 sent/received in the same radio frame, it rejects fabricated SIB messages.
- *Relay/replay attacks.* Since most of the parameters in the SIB1 and SIB2 messages are constant, our proposed solution uses the timestamp t as a nonce for generating non-deterministic signatures avoiding replay attacks. The UE identifies the freshness of any SIB2 message using the t and the Δ_t parameters of the message. With the relay/replay protection incorporated into our proposed PKI scheme, it reduces the attack window and raises the bar for the attackers to perform DNS redirection and phishing attacks [42].
- *Control plane message injection.* With our defense in place, a UE is able to identify fake base stations by verifying the authenticity of SIB messages, and never establishes the RRC layer connection to the fake base station. This thwarts the fake base station from injecting unauthenticated messages. The UE, therefore, will not expose its IMSI, downgrade to 3G/2G, or prevents location tracking.

However, the adversary can still sniff messages sent by the legitimate parties and can overhear IMSI in the `attach_request` and the `identity_response` messages. This type of IMSI catching attack is very hard to detect since the adversary does not leave any footprints. The upcoming 5G solves this problem by requiring UEs to encrypt its IMSI when sending `attach_request` or `identity_request` messages.

8 DISCUSSION

Precise time synchronization. If users want to enjoy our proposed relay protection mechanism, a precise time synchronization between a UE and a base stations is required which can be reliably achieved through GPS timer. Since GPS timer may induce privacy risks, we minimize such risks by disabling the GPS as soon as the UE authenticates a base station and connects to the core network. Since a UE in connected state does not always listen to SIB messages unless there is a change of tracking area due to mobility or changes of network information or emergency notifications, GPS will remain off for majority of the time.

Limitations of relay protection mechanism. In our approach, due to over approximation of Δ_{TU}, relay attacks are still possible. We, however, argue that our proposed technique raises the bar for the attackers. A more concrete solution would potentially require changes to the protocol (e.g., additional messages) and require precise estimation of constants that due to environmental interference and hardware configurations can change considerably. Similarly, if Δ_{TL} is under approximated (e.g., UE uses faster hardware than originally accounted for), it could lead to UEs failing to authenticate. This, however, can be easily circumvented by keeping the constants up to date with respect to modern devices.

Emergency call setup. According to the 3GPP standard [2], devices without SIM/USIM/eSIM do not perform any authentication to connect to the core network during emergency call setup. Since our proposed solution does not encrypt any SIB messages at all, emergency call can be setup in the usual way.

Legacy devices. To maintain backward compatibility, we include the additional fields (e.g., signatures) of SIB1 and SIB2 messages as non-critical extensions. Legacy devices can ignore these non-critical extensions without jeoperdizing connectivity and functionality.

9 RELATED WORK

Fake base station detection. To detect fake base stations acting as IMSI-catchers, Dabrowski et al. [19] propose to use stationary hardware units to scan frequency bands, collect cellular data and find anomalous communication patterns. This, however, has the limitation of requiring expensive hardware units and scalability. In addition, Borgaonkar et al. demonstrate that such signature-based fake base detection schemes are susceptible to new attack variants [35]. Dabrowski et al. [18] also look into detecting such fake devices using the operator side data and combine both client and operator side detections which, however, are vulnerable since the data is generated and analyzed after the cellular devices connect to the IMSI-catchers. Li et al. [29] use crowdsourced data to detect fake base stations that broadcast fake SMS messages to scam the victims. Though they have promising results in this very specific scenario, there exists two important limitations. First, there is little to no ground truth available for this type of attack. Second, they can only detect fake base stations with known communication patterns that broadcast fake SMS messages [29]. Ney et al. [32] propose to solve this problem using sensors mounted in vehicles. This solution comes with the benefit that no subscribers need to connect to such devices to create traces which otherwise could later serve as a basis to detect them and instead use the data collected by these sensors. This, however, suffers from the limitation that such sensors would be expensive to deploy and would cause the scalability issues.

Preventing exposure of IMSI. Khan et al. [26] propose a solution to conceal the IMSI using Identity Based Encryption and provide authentication. This, however, comes with the challenge of imposing computation overhead at the Home Subscriber Network since it would require a public-private key pair for each subscriber. Pseudonym-based IMSI concealment techniques [5, 27, 47] might prevent the exposure of IMSI; however, the attacker could still perform downgrade attacks, and thus expose the IMSI through 4G/3G.

In the current version of the 5G protocol, the subscriber identity is encrypted with the network's public key to avoid exposure [1]. Even without a downgrade attack, Hussain et. al [23] demonstrate an attack that significantly reduces the search space and allows for a brute force attack to reveal the subscriber's identity.

Mutual Authentication. The root cause of IMSI-catchers is the failure to authenticate the fake base station prior to connection. A common approach to this problem is a PKI-based solution that fully relies on certificates. A common theme in these solutions is that the core network acts as a CA and, in the process, signs certificates for every MME/AMF in the network [28, 50, 51]. These solutions, however, impose a significant computational overhead at the base station and induce high communication overhead due to the lack of optimizations in authenticating a broadcast message. Though this solution proves to be computationally feasible, the IMSI can be still exposed due to the failure of SIB message's authentication.

10 CONCLUSION AND FUTURE WORK

In this paper, we investigate cryptography-backed authentication mechanisms to prevent adversaries, from luring unsuspecting cellular devices to connect to malicious base stations. We accomplish this by enabling next generation cellular devices to authenticate the legitimacy of a base station, prior to connection. We overcome the constraints imposed by both the ecosystem and stakeholders, and design an optimized PKI scheme. We leverage precomputation-based digital signature generation algorithms and employ different domain-specific optimizations to address the trilemma imposed by digital signatures. We then implement our mechanism and observe that our authentication scheme with the best performing digital signature algorithm imposes moderate overhead in bytes (~**220** bytes) and minimal overhead connection time wise (~**28** ms), all while maintaining backwards compatibility.

In future, we will customize our proposed scheme for resource constrained cellular IoT devices and 5G ultra-reliable low latency communication (URLLC) protocols.

ACKNOWLEDGEMENT

We thank our Shepherd, Ravishankar Borgaonkar, and the anonymous reviewers for their valuable suggestions. This work was supported by NSF grant CNS-1657124, NSF grant CNS-1719369 and Intel as part of the NSF/Intel ICN-WEN program.

REFERENCES

[1] [n. d.]. *3GPP Release 15.* http://www.3gpp.org/release-15.

[2] [n. d.]. *5G; Security architecture and procedures for 5G System (3GPP TS 33.501 version 15.1.0 Release 15).*

[3] [n. d.]. *Cell-Site Simulators/IMSI Catchers.* https://www.eff.org/pages/cell-site-simulatorsimsi-catchers.

[4] [n. d.]. *Embedded SIM Remote Provisioning Architecture (Version 1.1).* https://www.gsma.com/iot/wp-content/uploads/2014/01/1.-GSMA-Embedded-SIM-Remote-Provisioning-Architecture-Version-1.1.pdf.

[5] [n. d.]. *Protecting IMSI and User Privacy in 5G Networks.* www.ericsson.com/res/docs/2016/protecting-imsi-and-user-privacy-in-5g-networks.pdf.

[6] [n. d.]. *Qualcomm Snapdragon 845 Mobile Platform.* https://www.qualcomm.com/media/documents/files/snapdragon-845-mobile-platform-product-brief.pdf.

[7] [n. d.]. *srsLTE.* https://github.com/srsLTE.

[8] [n. d.]. *srsUE.* https://github.com/srsLTE/srsUE.

[9] [n. d.]. *USRP B210.* https://www.ettus.com/product/details/UB210-KIT.

[10] Dare Abodunrin, Yoan Miche, and Silke Holtmanns. 2015. Some Dangers from 2G Networks Legacy Support and a Possible Mitigation. In *Communications and Network Security (CNS)*. 585–593.

[11] Iosif Androulidakis. 2011. Intercepting Mobile Phone Calls and Short Messages Using a GSM Tester. In *18th Conference on Computer Networks, Ustron, Poland*, Andrzej Kwiecień, Piotr Gaj, and Piotr Stera (Eds.). 281–288.

[12] Myrto Arapinis, Loretta Mancini, Eike Ritter, Mark Ryan, Nico Golde, Kevin Redon, and Ravishankar Borgaonkar. 2012. New Privacy Issues in Mobile Telephony: Fix and Verification. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security (CCS '12)*. ACM, 205–216.

[13] Myrto Arapinis, Loretta Ilaria Mancini, Eike Ritter, and Mark Ryan. 2014. Privacy through Pseudonymity in Mobile Telephony Systems. In *NDSS*.

[14] American Bankers Association et al. 1999. Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm. *American National Standard X* 9 (1999).

[15] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. 2003. Aggregate and verifiably encrypted signatures from bilinear maps. In *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 416–432.

[16] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. 2003. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. In *Advances in Cryptology — EUROCRYPT 2003*, Eli Biham (Ed.). 416–432.

[17] David Cooper, Stefan Santesson, Stephen Farrell, Sharon Boeyen, Russell Housley, and Tim Polk. 2008. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile.* Technical Report 5280. http://www.ietf.org/rfc/rfc5280.txt

[18] Adrian Dabrowski, George 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, Cham, 279–302.

[19] 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 (ACSAC '14)*. 246–255.

[20] Nico Golde, Kévin Redon, and Ravishankar Borgaonkar. 2012. Weaponizing Femtocells: The Effect of Rogue Devices on Mobile Telecommunications. In *Proceedings of the 19th Annual Network & Distributed System Security Symposium*.

[21] Nico Golde, Kévin Redon, and Jean-Pierre Seifert. 2013. Let Me Answer That for You: Exploiting Broadcast Information in Cellular Networks. In *Proceedings of the 22Nd USENIX Conference on Security*. 33–48.

[22] Syed Rafiul Hussain, Omar Chowdhury, Shagufta Mehnaz, and Elisa Bertino. 2018. LTEInspector: A Systematic Approach for Adversarial Testing of 4G LTE. In *25th Annual Network and Distributed System Security Symposium, NDSS, San Diego, CA, USA, February 18-21*.

[23] 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 *26th Annual Network and Distributed System Security Symposium, NDSS, San Diego, CA, USA, February 24-27, 2019*.

[24] 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.

[25] Jonathan Katz, Alfred J Menezes, Paul C Van Oorschot, and Scott A Vanstone. 1996. *Handbook of applied cryptography*. CRC press.

[26] Mohsin Khan and Valtteri Niemi. 2017. Concealing IMSI in 5G Network Using Identity Based Encryption. In *axXiv preprint arXiv:1708.01868*.

[27] Mohammed Shafiul Alam Khan and Chris J Mitchell. 2017. Trashing IMSI Catchers in Mobile Networks. In *Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks*. 207–218.

[28] Cheng-Chi Lee, I-En Liao, and Min-Shiang Hwang. 2009. An Extended Certificate-based Authentication and Security Protocol for Mobile Networks. In *Information Technology and Control*, Vol. 38. Issue 1.

[29] Zhenhua Li, Weiwei Wang, Christo Wilson, Jian Chen, Qian Chen, Taeho Jung, Lan Zhang, Kebin Liu, Xiangyang Li, and Yunhao Liu. 2017. FBS-Radar: Uncovering Fake Base Stations at Scale in the Wild. In *24th Annual Network and Distributed System Security Symposium, NDSS, San Diego, CA, USA*.

[30] Ulrike Meyer and Susanne Wetzel. 2004. A Man-in-the-Middle Attack on UMTS. In *Proceedings of the 3rd ACM Workshop on Wireless Security (WiSe '04)*. ACM, 90–97.

[31] Collin Mulliner, Nico Golde, and Jean-Pierre Seifert. 2011. SMS of Death: From Analyzing to Attacking Mobile Phones on a Large Scale. In *Proceedings of the 20th USENIX Conference on Security (SEC'11)*. USENIX Association, Berkeley, CA, USA, 24–24. http://dl.acm.org/citation.cfm?id=2028067.2028091

[32] Peter Ney, Ian Smith, Gabriel Cadamuro, and Tadayoshi Kohno. 2017. SeaGlass: Enabling City-Wide IMSI-Catcher Detection. In *Proceedings on Privacy Enhancing Technologies*. 39–56.

[33] Karsten Nohl. [n. d.]. *Mobile Self Defense.* https://events.ccc.de/congress/2014/Fahrplan/system/attachments/2493/original/Mobile_Self_Defense-Karsten_Nohl-31C3-v1.pdf

[34] Joseph Ooi and Nadia Neninger. 2015. IMSI Catchers and Mobile Security. (2015).

[35] Shinjo Park, Altaf Shaik, Ravishankar Borgaonkar, Andrew Martin, and Jean-Pierre Seifert. 2017. White-Stingray: Evaluating IMSI Catchers Detection Applications. In *11th USENIX Workshop on Offensive Technologies (WOOT 17)*. USENIX Association, Vancouver, BC. https://www.usenix.org/conference/woot17/workshop-program/presentation/park

[36] Adrian Perrig, Ran Canetti, J.D. Tygar, and Dawn Song. 2005. The TESLA Broadcast Authentication Protocol. In *Cryptobytes 5*.

[37] Roger Piqueras Jover. 2013. Security Attacks Against the Availability of LTE Mobility Networks: Overview and Research Directions. In *Wireless Personal Multimedia Communications (WPMC), 2013 16th International Symposium on*.

[38] Kasper Bonne Rasmussen and Srdjan Čapkun. 2010. Realization of RF Distance Bounding. In *Proceedings of the 19th USENIX Conference on Security (USENIX Security'10)*.

[39] Bradley Reaves, Logan Blue, Hadi Abdullah, Luis Vargas, Patrick Traynor, and Thomas Shrimpton. 2017. AuthentiCall: Efficient Identity and Content Authentication for Phone Calls. In *26th USENIX Security Symposium (USENIX Security 17)*. Vancouver, BC, 575–592.

[40] Ronald L Rivest, Adi Shamir, and Leonard Adleman. 1978. A Method for Obtaining Digital Signatures and Public-key Cryptosystems. *Commun. ACM* 21, 2 (1978), 120–126.

[41] David Rupprecht, Adrian Dabrowski, Thorsten Holz, Edgar Weippl, and Christina Pöpper. 2018. On Security Research Towards Future Mobile Network Generations. *IEEE Communications Surveys Tutorials* 20, 3 (2018), 2518–2542.

[42] David Rupprecht, Katharina Kohls, Thorsten Holz, and Christina Pöpper. 2019. Breaking LTE on Layer Two. In *IEEE Symposium on Security & Privacy (SP)*. IEEE.

[43] Stefan Santesson, Michael Myers, Rich Ankney, Ambarish Malpani, Slava Galperin, and Carlisle Adams. 2013. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP. *RFC* 6960 (2013), 1–41. https://doi.org/10.17487/RFC6960

[44] 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 *23nd Annual Network and Distributed System Security Symposium, NDSS, San Diego, CA, USA, February 21-24*.

[45] Patrick Traynor, Michael Lin, Machigar Ongtang, Vikhyath Rao, Trent Jaeger, Patrick McDaniel, and Thomas La Porta. 2009. On Cellular Botnets: Measuring the Impact of Malicious Devices on a Cellular Network Core. In *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS)*. 223–234.

[46] Patrick Traynor, Patrick McDaniel, and Thomas La Porta. 2007. On Attack Causality in Internet-connected Cellular Networks. In *Proceedings of 16th USENIX Security Symposium*. Article 21, 16 pages.

[47] Fabian van den Broek, Roel Verdult, and Joeri de Ruiter. 2015. Defeating IMSI Catchers. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security (CCS '15)*. ACM, 340–351.

[48] 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.

[49] 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 (Nov 2017), 2627–2639. https://doi.org/10.1109/TIFS.2017.2716911

[50] Xun Yi, Eiji Okamoto, and Kwok Yan Lam. 1998. An Optimized Protocol for Mobile Network Authentication and Security. In *ACM SIGMOBILE Mobile Computing and Communications Reviews*, Vol. 2. 37–39. Issue 3.

[51] Yuliang Zheng. 1996. An Authentication and Security Protocol for Mobile Computing. In *Mobile Communications*. Springer US, 249–267.

[52] Zhou Zhuang, Xiaoyu Ji, Taimin Zhang, Juchuan Zhang, Wenyuan Xu, Zhenhua Li, and Yunhao Liu. 2018. FBSleuth: Fake Base Station Forensics via Radio Frequency Fingerprinting. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security (ASIACCS '18)*. 261–272.