AAKA: An Anti-Tracking Cellular Authentication Scheme Leveraging Anonymous Credentials

Hexuan Yu*, Changlai Du*, Yang Xiao†, Angelos Keromytis‡, Chonggang Wang§, Robert Gazda§,
Y. Thomas Hou*, Wenjing Lou*

*Virginia Polytechnic Institute and State University
†University of Kentucky
‡Georgia Institute of Technology
§InterDigital Inc.

***Abstract*—Mobile tracking has long been a privacy problem, where the geographic data and timestamps gathered by mobile network operators (MNOs) are used to track the locations and movements of mobile subscribers. Additionally, selling the geolocation information of subscribers has become a lucrative business. Many mobile carriers have violated user privacy agreements by selling users' location history to third parties without user consent, exacerbating privacy issues related to mobile tracking and profiling. This paper presents AAKA, an *anonymous authentication and key agreement* scheme designed to protect against mobile tracking *by honest-but-curious MNOs*. AAKA leverages anonymous credentials and introduces a novel mobile authentication protocol that allows legitimate subscribers to access the network anonymously, without revealing their unique (real) IDs. It ensures the integrity of user credentials, preventing forgery, and ensures that connections made by the same user at different times cannot be linked. While the MNO alone cannot identify or profile a user, AAKA enables identification of a user under legal intervention, such as when the MNOs collaborate with an authorized law enforcement agency. Our design is compatible with the latest cellular architecture and SIM standardized by 3GPP, meeting 3GPP's fundamental security requirements for User Equipment (UE) authentication and key agreement processes. A comprehensive security analysis demonstrates the scheme's effectiveness. The evaluation shows that the scheme is practical, with a credential presentation generation taking $\sim$52 ms on a constrained host device equipped with a standard cellular SIM.**

I. Introduction

Mobile cellular technologies have seen rapid advancement and ubiquitous deployment in the past three decades. From 2G to 5G, mobile networks are providing voice, text, and general data services, with enhanced coverage, connectivity, and data rate. A mobile subscriber, typically through a mobile phone, can make phone calls and connect to the Internet pretty much from anywhere at any time.

Mobile Tracking as a Privacy Threat. Despite the unprecedented connectivity and mobile communication services, cellular technologies put the subscriber's location and data privacy at risk, leading to the problem of mobile tracking. Mobile carriers, or Mobile Network Operators (MNOs), can uniquely identify and locate a subscribing user during the latter's cellular access, creating tracking profiles on where the user has been, whom she has been with, and how the user has accessed the network for what service. The capability of an MNO to track and profile its users is a serious privacy concern to many. The MNO's user location data could be illegally sold to a third-party data broker or Location-based Service (LBS) provider by an unprofessional employee or leaked to criminals due to a cyber-attack [3, 4, 46, 52]. Since 2020, the four major carriers in the U.S., AT&T, Verizon, Sprint, and T-Mobile, have been fined for violating the Communications Act and the Federal Communications Commission's (FCC) regulations governing the privacy of subscribers' location information, which resulted in the disclosure of subscriber location records to unauthorized third parties without subscribers' consent [40–43]. The reality is that such location information can be easily obtained from MNOs by a motivated individual, let alone a state-level adversary who may have direct control over the MNOs in its jurisdiction.

Meanwhile, recent advancements in localization and positioning technology powered by mmWave are posed to allow MNOs to locate User Equipment (UE) with sub-meter precision [34, 49]. Next-generation (**NextG**) mobile networks will continue the trend established by 5G New Radio (NR) systems and strive for more precise localization with higher frequency ranges, wider bandwidths, and increasingly dense antenna arrays [16]. While offering subscribers greater availability and accessibility, finer-grained localization amplifies the privacy risk—the accurate real-time location information of a UE is now fully transparent to MNOs, sufficient to profile a specific subscriber (e.g., workplace, home address) through pattern analysis. The spike in the number of cellular-connected devices (e.g., smartphones, smartwatches, tablets, massive IoTs) does not help either. The increased connectivity of personal devices further exacerbates the privacy risk as it creates more correlated traces for linkage attacks.

The mobile tracking problem is further complicated by the dilemma between subscriber privacy and accountability. The location tracking data from MNOs, most commonly the Cell-site Location Information (CSLI), has long been used by law enforcement for providing evidence to criminal investigations [53, 58]. Such data-gathering capability can nonetheless help unscrupulous law enforcement or states to conduct illegal surveillance. Recently, 3GPP communities are standardizing the *Lawful Interception* (LI) [9, 10] process within cellular

* A previous version of this work was published in NDSS 2024. A typo of Pres in Fig. 3 has been corrected in this version.

architecture, which aims to regulate the process for a Law Enforcement Agency (LEA) to access mobile user data via dedicated LI interfaces. The 3GPP LI standards have identified a list of capabilities that an MNO has to have for national security purposes. Basically, with the involvement of the LEA and a valid warrant, MNO's LI interface should be able to return a target UE's metadata (e.g., identity, timestamp, location reporting) or the content of communications. Triggering any LI functions must require a warrant obtained from LEA as input. However, this is usually handled manually in practice, and there are no rigorous technical solutions for preventing MNOs themselves from obtaining the LI data without LEA's presence.

Our Contribution. We aim to address the aforementioned challenges of mobile tracking in cellular networks, namely, preserving identity privacy, minimizing the subscriber's location footprint against untrusted MNOs (i.e., *insider attack*), and preventing illegal surveillance without impacting the current cellular access and service model.

We recognize that the fundamental enabler of mobile tracking is the inherent capability of MNOs to consistently acquire a user's permanent identifier while servicing the user. Specifically, when a mobile device attempts to connect to the cellular network, the user Authentication and Key Agreement (AKA) process ensues, allowing both the user's Home Network (HN) and Serving Network (SN) to gain real-time knowledge of the user's mobile access.

Therefore, we propose an efficient and backward-compatible Anonymous Authentication and Key Agreement scheme, dubbed AAKA, enabling a mobile user to access services without revealing her permanent identity and thus resist tracking. AAKA achieves anonymous access authentication by leveraging the anonymous credential (AC) techniques, and utilizes the cellular **SIM** (Subscriber Identity Module), standardized by 3GPP [7], as the tamper-proof secure element to store UE's credentials. AAKA consists of two sub-protocols:

1) a *subscription credential issuance protocol*, where HN issues UE a verifiable AC based on its subscription status;
2) a *presentation and verification* protocol, where UE derives a one-time verifiable presentation from the AC which hides the permanent UE identifier and presents it to SN, fulfilling authentication and key agreement anonymously.

The proposed protocols protect the subscriber's identity and location privacy by ensuring ANONYMITY and UNLINKABILITY.

In addition to effectively countering mobile tracking threats, our design guarantees protection against common security threats like replay attacks, impersonation, and eavesdropping, as specified in the latest security requirements for 5G networks by 3GPP [11]. While preserving subscriber privacy by default, AAKA also strives to provide a transparent and computationally secure lawful de-anonymization method to uphold user ACCOUNTABILITY under a valid search warrant (to support LI functions).

In summary, the proposed AAKA accomplishes the following:

- **Anti-Tracking Privacy Enhancement.** AAKA leverages ACs and *zero-knowledge proofs* in the credential presentation, allowing a UE to *selectively disclose* its verifiable attributes and gain access to mobile networks anonymously. Credential presentations generated by different subscribers are indistinguishable, and multiple accesses associated with a single subscriber remain unlinkable.
- **Anti-Counterfeiting Protection.** The credentials used in AAKA are unforgeable and non-transferable. This is ensured by the underlying crypto primitives adopted, the hardware-level protection available at SIM, and the secure Over-the-Air (**OTA**) SIM provisioning protocol.
- **Lawful De-anonymization.** To enable LI while protecting user privacy by default, AAKA supports "*Geofence Search Warrant*" (also known as "Reverse Location Search Warrant") through a novel *Identity Escrow* scheme (Section V-E), which allows the MNOs (include both SN and HN) and LEA to work together to de-anonymize the user identities within a specific geographical area and time frame as mandated by the warrant. We provide a cryptographic guarantee that UE de-anonymization can only be accomplished with the presence of both MNOs and LEA. Neither a curious MNO nor an unauthorized LEA alone can de-anonymize any UE, thus preventing the misuse of subscribers' data by either party.
- **Non-interactive Roaming Support.** The current primary mutual authentication and key agreement process between UE and MNO (i.e., 5G-AKA or EAP-AKA'[1]) is an interactive Challenge-Response process that requires the involvement of HN during each UE registration process. Regarding roaming, the foreign SN needs to establish communication with the HN of the UE that initiated the registration. As a performance enhancement over the latest AKA scheme (i.e., 5G-AKA), AAKA enables an SN to independently authenticate an unknown UE during general roaming scenarios, without interacting with the UE's HN. This leads to additional savings in communications and is made possible via our *non-designated* credential verifier method (Section V-C).
- **Compatibility and Practicality.** AAKA utilizes the standard SIM and existing cellular infrastructure without introducing new entities to the mobile networks. Our optimized anonymous credential protocol is *pairing-free* on the UE side, and all the cryptographic computations for UE are feasible with the latest cellular SIMs. The experimental result further demonstrates the practicality of the computational costs on UEs, as a credential presentation generation takes ∼52 ms on a constrained host device. The overall UE registration process adds ∼60 ms compared to that of 5G-AKA.

Outline. The rest of this paper is organized as follows. In Section II we briefly explain how the identity and mobility events of a particular UE are linked and tracked by MNOs within the present cellular networks. Section III presents the threat model and a high-level system overview. Section IV introduces the cryptographic building blocks and primitives. Section V elaborates our privacy-preserving scheme, while security analysis is covered in VI. The experimental results are reported and discussed in Section VII. Section VIII concludes the paper and discusses future works.

[1] EAP stands for "Extensible Authentication Protocol", and it only slightly differs in key derivation comparing to 5G-AKA. Since 5G-AKA is more prevalent in practice, we use it as an example in all illustrations pertaining to the current primary authentication.

TABLE I: 5G Abbreviations and Definitions

Abbreviation	Definition
AMF	Access and Mobility Management Function
ADMF	Administration Function
ARPF	Authentication Credential Repository and Processing Function
AUSF	Authentication Server Function
GUTI	Global Unique Temporary Identifier
HN	Home Network
LEMF	Law Enforcement Monitoring Facility
LMF	Location Management Function
LI	Lawful Interception
LICF	LI Control Function
LIPF	LI Provisioning Function
MCC	Mobile Country Code
MDF	Mediation and Delivery Function
MNC	Mobile Network Code
MSIN	Mobile Subscriber Identification Number
OTA	Over-The-Air
RA	Registration Area
SEAF	Security Anchor Function
SN	Serving Network
SUCI	Subscription Concealed Identifier
SUPI	Subscription Permanent Identifier
TA	Tracking Area
UE	User Equipment
USIM	Universal Subscriber Identity Module

II. Background and Related Work

In order to better comprehend the root cause of the mobile tracking problem, we provide a succinct background on the subscription credentials, authentication, and UE registration processes of the incumbent mobile networks in this section. Glossaries that appear in this paper are listed in Table I.

A. Subscription Credentials and Identifiers

In mobile networks, **USIM** (Universal Subscriber Identity Module) refers to the **SIM** application running on a hardware chip called **UICC** (Universal Integrated Circuit Card)[2]. **ME** (Mobile Equipment), e.g., a smartphone, an eSIM-enabled IoT device, etc., comprises a **UE** together with the cellular (e)SIM.

Subscription Credentials in the current mobile networks consist of the **SUPI**[3] (Subscription Permanent Identifier) and long-term secret keys (e.g., subscriber permanent key k), which, when put together, can uniquely identify a USIM and facilitate mutual authentication between UE and the core network. Note that, an **IMSI-type SUPI** is equivalent to the primary identifier **IMSI** (International Mobile Subscriber Identity) that was used before 5G. As SIM is a tamper-resistant secure hardware, the subscription credentials are securely stored in USIM on the UE side, and ARPF (Authentication Credential Repository and Processing Function) on the HN side, respectively [11].

SUPI contains the *HN Identifier*, i.e., mobile network code (**MNC**) and mobile country code (**MCC**), which is used to identify the UE's HN, and a Mobile Subscriber Identification Number (**MSIN**), which serves to identify particular UE within one MNO.

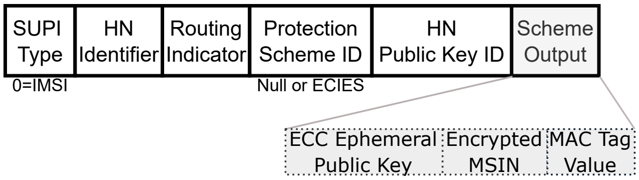

Fig. 1: SUCI Composition

The confidential subscription credentials and the public key of HN (i.e., pk_{HN}) can be securely stored in USIM by HN through the OTA provisioning process. The Long Term Key Update Process (LTKUP) service of the UDM **OTA** server residing in the HN can replace the subscription credentials within USIM if necessary, through dedicated OTA interfaces [5]. For instance, when key exposure is detected, the LTKUP service will be activated to update the k through remote APDU (Application Protocol Data Unit) commands.

Subscription Concealed Identifier (**SUCI**) [6]), an *encrypted* version of SUPI, has been introduced since 5G to protect UE privacy and prevent outsider attacks. SUCI composition is shown in Fig. 1.

As defined in TS 33.501 [11], when a protection scheme is enabled, UE utilizes SUCI to identify itself when transmitting over ngRAN rather than SUPI. In special circumstances, such as an emergency call session where authentication can be bypassed, the *Null* scheme is chosen, and SUPI will be transmitted unencrypted.

On the UE side, SUCI calculations can be done either by USIM or the ME according to HN's indication. SUCI encrypts only the MSIN field of the original SUPI utilizing ECIES (Elliptic Curve Integrated Encryption Scheme) and does not conceal the routing-related information, including *HN Identifier* and *Routing Indicator*. The keys involved in the ECIES encryption process are pk_{HN}, and an ephemeral EC secret key generated by UE. Only HN, who possesses sk_{HN}, can decrypt the SUCI into SUPI.

B. UE Authentication and Registration

We take a general roaming scenario as an example. Before attaching to the cellular network within a foreign SN region, UE is required to perform a mandatory AKA process (details are provided in Appendix A) to mutually authenticate between herself and SN, and establish a secured anchor key (i.e., K_{SEAF}) used in the subsequent security procedures. The authentication vectors used during an AKA process are essentially a set of challenge-response messages spawned from the pre-provisioned confidential materials (e.g., k and pk_{HN}) jointly with several fixed KDFs (Key Derivation Functions, defined in Annex A TS 33.501 [11]) that known to both USIM and HN. SN can only authenticate UE with the assistance of UE's HN, by acting as an intermediary and relaying these authentication vectors between them. After successful authentication, HN will inform SN of the SUPI of UE, and provide SN with the anchor key K_{SEAF} that is generated by the AUSF (Authentication Server Function) of HN. UE can independently derive the K_{SEAF} through the KDFs in USIM, yielding an implicit key agreement.

[2] For the sake of simplicity, we will not specifically differentiate between USIM, (e)UICC, and (e)SIM in the descriptions that follow.

[3] SUPI is an identifier used within MNOs, while the phone number, Mobile Subscriber ISDN Number (MSIDDN), is not required for network operation in reality.

After a successful AKA procedure, the corresponding **AMF** (Access and Mobility Management Function) in the SN will assign this UE a temporary identifier, **GUTI** (Global Unique Temporary Identifier), indicating a successful UE registration. This GUTI can be used within the Registration Area (RA) of this AMF; hence, the UE does not need to undertake another registration process if she is still within the same RA and possesses a valid GUTI. One RA could consist of more than one Tracking Area (TA, i.e., a group of cells), hence reducing the frequency of UE registration.

Even under *Idle* mode, UE will be periodically paged, and the location is reported by the ng-RAN as the cell identity. AMF receives and transmits the location-related metadata (e.g., timestamp) between ngRAN and the Location Management Function (LMF). Once connected to the mobile networks, UE location is precisely recorded by the dedicated NFs, while a centimeter-level localization is anticipated in the near future, the trajectory of the UE's movement is becoming increasingly clear to the MNOs.

C. Related Work

Before 5G, law enforcement and threat actors alike have leveraged a special device called *IMSI-Catcher*, or *Stingray*, to acquire the permanent identity (i.e., IMSI) of a UE in the wild [55]. IMSI catching is essentially a man-in-the-middle attack,

enabled by the fact that IMSI was sent in plaintext during the 2G, 3G, and 4G eras. IMSI catching was pervasive and easy to carry out by both legal and illegal parties due to its low cost [55]. Although 5G has started to protect SUPI and transmit it in an encrypted manner to thwart IMSI catching attacks [11, 14, 64], literature shows that IMSI remains vulnerable to exposure under various attacks, such as forcing UE to downgrade to 2G LTE and leaving IMSI unencrypted yet again (i.e., Bidding-Down attacks) [50, 56]. Meanwhile, an ongoing debate persists within the legal community on how to resolve the conflict between collecting location-tracking data for investigation or surveillance and respecting users' expectations of location privacy [30].

Most existing works of preserving subscribers' identity and location privacy under 5G and beyond settings are against *outsiders*. For example, AKA+ [51] and AKA' [64] optimized the current 5G-AKA protocol to resist linkability attacks conducted by an active outsider (e.g., linkage brought by a stateful sequence number synchronization); Hong et al. [48] revealed that infrequent refreshing renders GUTI a quasi-permanent identifier and reintroduces linkability risks, and presented an unpredictable GUTI reallocation mechanism to protect UE location privacy. Budykho et al. [21] also demonstrated that reusing GUTI across different sessions during the 5G handover procedure can bring trackability risk. Du et al. investigated the UE privacy threats posed by malicious MNOs and posited potential directions [38], however, there are only a few solutions in the existing literature. ZipPhone [62] modeled and quantified the location predictability and trajectory attacks while assuming a subscriber has the ability to switch ephemeral IMSI at her will, however, a realistic solution of how IMSI can be updated on a subscriber side is not discussed. PGPP [59] analyzed how the TAs assigned to each UE can be randomized to resist outsider attacks; besides, they proposed to nullify permanent identifiers to resist malicious MNOs, by replacing SUPI with a token (blind signature) for fulfilling UE authentication. Nevertheless, the authentication relies on a gateway as the trusted third party, which is challenging to put into practice, and requires modifications of current cellular architecture, in addition, the SUPI nullification strategy brings new challenges of call and text routing.

III. System Model and AAKA Overview

In this section, we describe the participant model and security assumptions of the anonymous and accountable mobile network access framework, followed by an overview of AAKA.

A. Participant Model

We consider three participant types in our system: **UE**, **MNO**, and **LEA**.

UE embodies the user who wishes to access the mobile network. UE consists of the Mobile Equipment (ME) and the SIM. ME can be any mobile device (e.g., smartphone, IoT device) that has the capability to embed a SIM. Aside from the current subscription materials (e.g., SUPI and k), SIM cards also hold verifiable anonymous credentials that enable users to access an MNO's service without disclosing their SUPI.

MNO is either (1) a **HN**, issuing subscription credentials to its subscribing UE; or (2) a **SN**, providing network access to a UE after verifying the UE's credentials. UE's identity is only visible to its HN during the credential issuance procedure, while all network access is anonymous to either the HN or SN.

LEA is the lawful entity that performs lawful de-anonymization of a targeted anonymous UE. LEA is crucial in upholding user accountability; however, it needs to do so in collaboration with the MNOs and possess a valid legal warrant, such as a "Geofence Search Warrant".

B. Threat Model

1) Security Assumptions of Network Entities: The primary adversaries we consider here are the *honest-but-curious* **MNO**s that may compromise UE's privacy by profiling and abusing their customers' location information. Nevertheless, they will honestly adhere to the prescribed protocols within cellular networks, such as provisioning authentic credentials with accurate attributes into the SIM and correctly executing the authentication procedures as required.

Our scheme is designed to be resistant to IMSI-catching attacks, thereby preventing unauthorized LEAs alone from deliberately identifying, monitoring, and tracing individuals within a specific area. Note that our threat model does not consider potential collusion between LEA and MNOs.

SIM/USIM, as a tamper-resistant secure element (SE), serves as the *root-of-trust* for the mobile device. We assume that UE-specific confidential materials (such as the long-term key k and **SUPI**) provisioned by HN are stored securely within the SIM, possessing inherent resistance to extraction, duplication, and manipulation.

ME is regarded as a *semi-trusted* entity, for which the security assumption aligns with the current mobile networks. It can

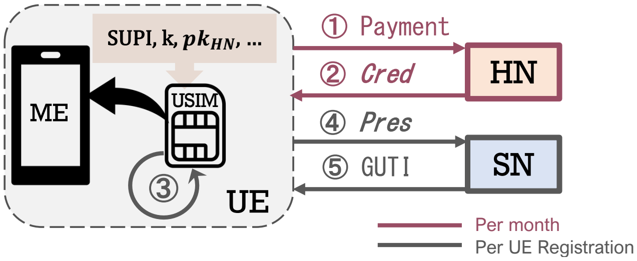

Fig. 2: System Entities and Simplified Workflow

be optionally utilized to compute non-critical cryptographic tasks offloaded from the embedded SIM. However, these computing and storage tasks will be carefully split to avoid being misused by a dishonest ME (discussed in Section VI-B).

2) Security Assumptions of Communication Channel: We follow the 3GPP security requirements [11] when considering channel conditions within mobile networks. We take into account both passive and active attackers in our defense. A *passive attacker* may eavesdrop on the messages exchanged between UE and SN before establishing a secure channel. Session keys will safeguard messages after the key agreement. An *active attacker* intercepts and manipulates messages transferred between UE and SN. An attacker, for example, could send a previously intercepted communication to an SN. Alternatively, an attacker may launch an IMSI-Catcher attack by impersonating a fake base station, etc. We will discuss how AAKA protects against various attacks while maintaining confidentiality, integrity, and authenticity over insecure channels.

C. AAKA Overview

The high-level workflow of AAKA is shown in Fig. 2. Our scheme emphasizes the process of anonymous authentication and key agreement. To simplify protocol illustration, the example workflow assumes that subscribers are enrolling in a traditional monthly prepaid unlimited plan. It's worth noting that AAKA potentially supports usage-based billing, which is covered in Section VIII. Besides, the base station (i.e., RAN), an intermediary that relays communications between UE and the core network (i.e., HN, SN), has been omitted from the subsequent explanations for the sake of brevity.

In AAKA, HN only needs to issue UE a subscription credential Cred (②) once a month (synchronous with payment interval ①) via the OTA SIM provisioning process. To obtain a valid Cred from HN, the subscriber must have paid for the upcoming month in step ① (via any conventional payment method). HN will prepare customized attributes (e.g., date of expiration, SUPI, etc.) and trigger the credential issuance phase ② after verifying the payment record.

By leveraging zero-knowledge proof techniques, UE is able to efficiently derive a *one-time* verifiable presentation Pres from the obtained credential Cred (③). While attempting to register with a SN (step ④), UE will first authenticate SN's identity, and deliver the Pres to SN to authenticate herself. The verifiable presentation Pres will neither reveal UE's identity (i.e., **SUPI**, **SUCI**) nor any other permanent metadata (e.g., permanent keys), that could cause linkability complications.

The steps ④ and ⑤ in Fig. 2 depict **UE Registration** during the *roaming* scenario. We take it as an example since it is more general than the *non-roaming* case (i.e., **SN = HN**) in typical cellular networks. Our technique is substantially simpler than the current primary authentication protocol (i.e., 5G-AKA) in the case of roaming, given that the SN does not need to contact the UE's HN in order to perform authentication and key agreement.

After verifying the Pres successfully, SN will register this UE to the associated **AMF** (step ⑤), and assign her a fresh **GUTI** for the upcoming sessions. A shared session key will also be produced during this step to secure subsequent communications.

IV. Cryptography Preliminaries

A. Bilinear Groups

A variety of anonymous credential schemes [17, 24, 25, 27, 29, 57] utilized pairing-based cryptography to solve DDH problems through the bilinear mapping properties [18–20]:

Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ be three bilinear cyclic groups with the same prime order p. g_1 and g_2 are the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. $\psi(g_2) = g_1$, and ψ is an isomorphism form between $\mathbb{G}_2 \rightarrow \mathbb{G}_1$. e denotes the bilinear mapping relationship $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, where $|\mathbb{G}_1| = |\mathbb{G}_2| = |\mathbb{G}_T| = p$. The bilinear map should satisfy the following properties: (1) for all $u \in \mathbb{G}_1, v \in \mathbb{G}_2$, and $a, b \in \mathbb{Z}_p^*$, $e\left(u^a, v^b\right) = e(u, v)^{ab}$ (**Bilinearity**). (2) $e(g_1, g_2) \neq 1$. ($e(g_1, g_2)$ is the generator for $\mathbb{G}_T$) (**Non-singular**) (3) e can be efficiently computed.

Due to implementation efficiency considerations, our protocols will be presented in asymmetric Type-3 settings [47]. In Type-3 setting, an efficient computable homomorphism ψ between $\mathbb{G}_1$ and $\mathbb{G}_2$ does not exist.

B. Proof of Knowledge

The concept of zero-knowledge came out in the 1980s [44]. A zero-knowledge proof of knowledge, or ZKP, is essentially a protocol where a prover convinces a verifier that he has knowledge of certain secrets without actually revealing them. Schnorr's identification protocol [60] is considered the first ZKP protocol, in which the prover can prove knowledge of a secure $x \in \mathbb{Z}_p$ to the verifier with respect to her public key $h := g^x \mod p$. The original Schnorr's protocol is a three-round interactive, honest-verifier protocol, by applying Fiat-Shamir heuristic [45] under the Random Oracle Model (ROM) [61], it can be transformed into a non-interactive zero-knowledge protocol (NIZK) and secure against arbitrary cheating verifiers. In the following sections, we use the zero-knowledge proof notation introduced by Camenisch and Stadler [28] for proving knowledge of discrete logarithms and concurrent statements of discrete logarithms during miscellaneous Σ-protocol [33] construction. An example ZKP is expressed as:

$$\pi \in PK\{(\alpha, \beta, \gamma) : y_1 = g^{\alpha} h^{\beta} \wedge y_2 = g^{\gamma}\} \qquad (1)$$

It denotes: a zero-knowledge proof of knowledge of α, β, γ such that $y_1 = g^{\alpha} h^{\beta}$ and $y_2 = g^{\gamma}$ holds, where g and h are elements of group $\mathbb{G}$ with prime order p. The parameters before the colon denote the secret to be proved, while all the remaining parameters are known to the verifier. By convention, π outputs a Boolean number to denote if the proof succeeds or not. Signature proof of knowledge (SPK), as formally defined by Chase et al. in [31], is a variant of ZKP, and is interchangeable with the notation PK when the secret to be proved is a valid signature.

C. Boneh-Boyen Signature

Boneh-Boyen signature (BBS) [15] is a short signature scheme existentially unforgeable against a non-adaptive chosen message attack under the q-SDH assumption. Given a signer's secret key x, signer's public key $X = g_2^x$ and two random generators $g_1 \in \mathbb{G}_1$, $g_2 \in \mathbb{G}_2$, the signature of a message $m \in \mathbb{Z}_p$ is generated by computing $\sigma = g_1^{1/(x+m)}$. The validity of the signature σ over m can be checked through a bilinear pairing e, i.e., $e(\sigma, Xg_2^m) \overset{?}{=} e(g_1, g_2)$.

Camenisch et al. [23] proposed an efficient way of proving knowledge of a Boneh-Boyen signature without revealing the message m: the prover randomly generates a blinding factor $r \in \mathbb{Z}_p$, sets $\sigma^{'} = \sigma^r$, and $\bar{\sigma} = \sigma^{'-m} g_1^r$. Excepting verifying $\sigma^{'} \neq 1 \in \mathbb{G}_1$ and the bilinear pairing $e(\bar{\sigma}, g_2) \overset{?}{=} e(\sigma^{'}, X)$, the verifier needs to verify a zero-knowledge proof:

$$\pi \in SPK\{(m, r) : \bar{\sigma} = \sigma^{'-m} g_1^r\} \qquad (2)$$

By introducing a proper auxiliary function, this construction eliminates pairing operations on the prover side. Inspired by this technique, a similar mechanism is applied in our scheme for the reality that the prover (i.e., UE) is typically a constrained device, whereas the verifiers (i.e., MNOs) are considered to have powerful computing servers. Thus, avoiding expensive pairing operations at the UE side can significantly improve protocol efficiency and feasibility.

D. Keyed-Verification Anonymous Credentials

Keyed-Verification Anonymous Credentials (KVAC), proposed by Chase et al. [32], is a type of AC that allows for more efficient composition. The KVAC is constructed using an algebraic message authentication code (MAC), which is a MAC formed using group operations as opposed to the more conventional hash or block cipher. It enables convenient integration with zero-knowledge proofs of discrete logarithms, offering great versatility and efficiency during issuing and proving possession of credentials.

Our design utilizes a KVAC construction proposed by Camenisch et al. [22], which incorporates the Boneh-Boyen signature. We refer to this scheme as **BBS-KVAC**, it essentially constructs algebraic MACs utilizing Boneh-Boyen signatures for a vector of messages $\vec{m} = (m_1, m_2, ..., m_n)$, with $m_i \in \mathbb{Z}_p^*$. The group $g \in \mathbb{G}$ is of prime order q, and $par = (\mathbb{G}, g, p)$ are public parameters. During the setup phase, the issuer generates a vector of issuing keys by randomly choosing $x_i \in \mathbb{Z}_p^*$ for $i = (0, 1, ..., n)$. The secret issuing keys are $ik = (x_0, x_1, ..., x_n)$, and public keys are $par = (X_0, X_1, ..., X_n)$ with $X_i = g^{x_i}$. The Boneh-Boyen signature of $\vec{m}$ is generated as $\sigma = g^{\frac{1}{x_0 + \sum_{i=1}^n m_i x_i}}$, with auxiliary parameters $\sigma_i = \sigma^{x_i}$ for $i = (0, 1, ..., n)$. The resulting MACs are: $(\sigma, \sigma_0, \sigma_1, ..., \sigma_n)$. To verify this vector of MACs (or a vector of Boneh-Boyen signatures), given $(ik, \vec{m}, \sigma)$, check if $g^{x_0 + \sum_{i=1}^n m_i x_i} = \sigma$.

TABLE II: Cryptographic Notations

Notation	Definition
$\mathbb{G}_1$,$\mathbb{G}_2$,$\mathbb{G}_T$	Multiplicative groups of prime order p
g_1,g_2	Generators of $\mathbb{G}_1$,$\mathbb{G}_2$
x_i	Issuance/Signing Keys of HN
X_i	Public Verification Keys of HN
α,β	Random Challenges
U,V	Secret keys for Diffie-Hellman Key Agreement
K_s	Ephemeral Diffie-Hellman Shared Key
$\vec{m}$	A vector of attributes
r,a,b	Blinding factors
σ	Boneh-Boyen Signatures
$\hat{\sigma_i}$	Blinded Boneh-Boyen Signatures
h	Public Key of LEA, i.e., pk_{LE}
(c_1, c_2)	ElGamal Ciphertext
$\bar{\sigma}$,A,B,b,$a^{'}$,$b^{'}$,y_i,$y_i^{'}$	Auxiliary components for ZKP
C	Challenge for ZKP

Lastly, $(m_1, m_2, ..., m_n)$ can be selectively disclosed per the ZKP scheme of Eq. (2), resulting in an AC scheme.

Theorem 1. (UF-CMVA SECURITY) *This BBS-based keyed-verification anonymous credential scheme (BBS-KVAC) is* ***uf-cmva-secure*** *for all the probabilistic polynomial-time (PPT) adversaries following the security definitions defined by Dodis et al. [36] and Chase et al. [32], i.e., it is* ***unforgeable*** *under chosen message and chosen verification queries attack under the n-Strong Computational Diffie-Hellman Inversion Problem (SCDHI) assumption [22] (covered in Appendix B).*

However, existing KVAC schemes [22, 32] have the limitation of being dependent on the designated verifier assumption, which states that either the verifier is also the credential issuer or the verifier knows the signing/issuing keys. This constraint renders it impractical in situations involving multiple authorities who are unable to share issuing keys ik with each other. For example, Verizon and AT&T can only issue credentials to their own subscribers. In order to circumvent this limitation, we present in Sections V-B and V-C, as a matter of independent interest, an enhancement to the existing KVAC schemes designed to accommodate *non-designated verifier* scenarios, i.e., a verifier without knowledge of the ik can still verify the credentials without affecting the prover's privacy guarantees.

E. ElGamal Encryption

The ElGamal cryptosystem [39] is semantically secure under the Decisional Diffie-Hellman (DDH) assumption. Given a cyclic group $\mathbb{G}$ of prime order p with generator $g \in \mathbb{G}$. Randomly take $x \in \mathbb{Z}_p$ as the secret key sk, sets public key $pk : h = g^x$. For a given message m, the encryption procedure runs as follows: Take a random value $r \in \mathbb{Z}_p$, $s = g^r$, and sets $c_1 = g^r$, $c_2 = mh^r$. The resulting ciphertext is (c_1, c_2). Given the public parameters $(\mathbb{G}, g, p)$ and secret key x, the decryption can be done by computing $m = c_2/c_1^x$. As a family of DH-based cryptosystems, ElGamal and its variants have been integrated into many advanced cryptographic protocols due to its homomorphism, e.g., distributed key generation (DKG) and threshold cryptosystems [35].

V. Protocol Detail

Our scheme consists of two sub-protocols: **Credential Issuance** (①②③ in Fig. 2), **Presentation and Verification** (④⑤ in Fig. 2). This section details the two sub-protocols and how they achieve the aforementioned design goals of AAKA.

A. System Setup

We first assume all the MNOs have agreed on a universal credential format. During the initial phase (e.g., before HN sends the subscriber with a physical SIM or remotely provisions eSIM profiles to the subscriber's SIM), the following parameters should be securely provisioned into the root-of-trust **SIM/USIM**: (1) the permanent identifier **SUPI**, (2) the permanent k that SIM shares with HN, (3) public key of LEA, i.e., pk_{LE}, which can be used for activating LI related network functions, such as **MDF** (Mediation and Delivery Function), (4) public key of HN, pk_{HN}, and (5) pars, public parameters used to verify issued credentials Cred and presentations $Pres$. These materials are usually permanent but can be securely replaced via an LTKUP procedure defined by TR 33.935 [5] under specific circumstances, such as a carrier database leakage or a hardware compromise. In addition to the above parameters, HN exclusively stores the secret issuance/signing key ik associated with the public verifying parameters $pars$.

The generation of **SUPI** and k is omitted in our illustration since they will be produced through the standardized solution defined in [11]. Formally, the system setup is as follows:

Setup. $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, p) \leftarrow \mathsf{Setup}(1^\lambda)$.
On input a security parameter 1^λ, outputs public group parameters pars : $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, p)$, where p is a λ-bit prime number.

Key Generation. $(par, ik, h, sk_{LE}) \leftarrow \mathsf{KeyGen}(\mathsf{pars})$.
KeyGen is a PPT key generation algorithm. HN chooses secret $x_i \in \mathbb{Z}_p^*$ for $i = (0, 1, ..., n)$ ($n = 4$ in our scheme as we require a Cred embedded with 4 attributes, see Section V-B), and output public verification parameters $pars = (X_0, X_1, ..., X_n)$ with $X_i = g_2^{x_i}$. We denote the issuing key or signing key as $ik = (x_0, x_1, ..., x_n)$. Given $\mathbb{Z}_p^*$, LEA outputs a pair of ElGamal asymmetric keys $(pk_{LE}, sk_{LE}) = (g_1^{sk_{LE}}, sk_{LE})$ as in Section IV-E. To simplify our cryptographic notations, we will denote $h = pk_{LE} (= g_1^{sk_{LE}})$ in the following sections.

B. Credential Issuance

Credential issuance is initiated by HN and occurs immediately after the subscriber successfully makes a payment. Upon receipt of the payment, HN will prepare the corresponding attributes for this UE. There are 4 attributes (m_1, m_2, m_3, m_4) embedded in Cred:

1) m_1, a boolean number indicates *subscription activity status*;
2) m_2, *expiration date*, which is usually aligned with the payment cycle;
3) m_3, *HN ID* (i.e., MCC + MNC), to indicate which HN the UE belongs to. This is used to inform the SN which public key and $pars$ he should refer to when authenticating the UE;
4) m_4, **MSIN**, as mentioned in Section II, is the only unique field of **SUPI**, which is used to uniquely identify a subscriber within the HN.

This sub-protocol is defined by two algorithms Issue and Obtain:

Credential Issuance. $(\mathsf{Cred}, \pi) \leftarrow \mathsf{Issue}(ik, \vec{m})$.
The attributes vector $\vec{m} = (m_1, m_2, m_3, m_4)$ as mentioned above. On inputting the issuing key $ik = (x_0, x_1, ..., x_4)$ and the $\vec{m}$, HN computes a signature $\sigma = g_1^{\frac{1}{x_0 + \Sigma_{i=1}^4 m_i x_i}}$ and generates a set of auxiliary signatures $\sigma_i = \sigma^{x_i}$ for $i = (0, 1, ..., 4)$, which will be used by UE during presentation generation phase. The resulting Cred is $(\vec{m}, \sigma, \{\sigma_i\}_{i=0}^4)$. Besides, HN should append a proof π_0 to prove that: (1) the auxiliary parameters are correctly formed through his secret signing key ik; (2) he has knowledge of the issuing key ik that corresponds to the public verification parameters $pars$.

$$\pi_0 \in ZKP\{(\{x\}_{i=0}^4) : \bigwedge_{i=0}^{4} \sigma_i = \sigma^{x_4} \wedge X_i = g_2^{x_i}\} \quad (3)$$

As described earlier, the subscription credentials can only be provisioned into USIM by HN via the dedicated 5G OTA remote provisioning channel. Thus (Cred, π_0) can be securely sent to USIM.

Credential Obtaining.
$(\mathsf{Cred}) \leftarrow \mathsf{Obtain}(par, \vec{m}, \sigma, \{\sigma_i\}_{i=0}^4, \pi_0)$.
In step $\underline{4}$, UE parses the received Cred as $(\vec{m}, \sigma, \{\sigma_i\}_{i=0}^4, \pi_0)$, before accepting it as valid, **USIM** (or UE) needs to:

1) verify that the provided σ, $\{\sigma_i\}_{i=0}^4$ correctly signed the desired attributes $\vec{m}$ by checking if $\sigma_0 \prod_{i=0}^4 \sigma_i^{m_i} = g_1$ holds;
2) verify that the (auxiliary) signatures $\{\sigma_i\}_{i=0}^4$ are indeed produced from the HN's issuing key ik;
3) verify that the issuing key ik used here is authentic with respect to the public verification parameters $pars$.

The properties 1) and 2) are confirmed simultaneously by non-interactively computing over the SPK π_0 using the classic AND–composition Σ–protocol. The π_0 here is essentially to prove knowledge of a set of discrete log $ik = (x_0, x_1, ..., x_4)$ w.r.t. σ and public parameter $pars$, such that $\sigma_i = \sigma^{x_i}$ and $X_i = g_2^{x_i}$ are true for $i = (0, 1, .., 4)$. The detailed computation for verifying π_0 is provided in Appendix D.

If the three properties mentioned above are successfully validated, UE will accept the Cred and store it on USIM. Now USIM possesses a valid Cred along with other essential materials, as shown in the left shaded box in Fig. 3.

C. Presentation and Verification

The *Presentation and Verification* protocol is an alternative to the existing $\mathsf{5G\text{-}AKA}$ process, aims to perform mutual authentication and key agreement between UE and SN, and register UE to the corresponding AMF under SN. We depict a *roaming* case in which UE tends to register herself to a foreign SN anonymously. Note that in situations where maintaining UE privacy is not required (e.g., emergency call sessions), SUPI shall be sent in plaintext (i.e., Null scheme mode in Fig.1) without executing the following steps.

As shown in Fig. 3, UE consists of a root-of-trust USIM and a semi-trusted ME, and the established shared key K_s must be kept secret from the ME throughout the protocol. The pk_{SN}

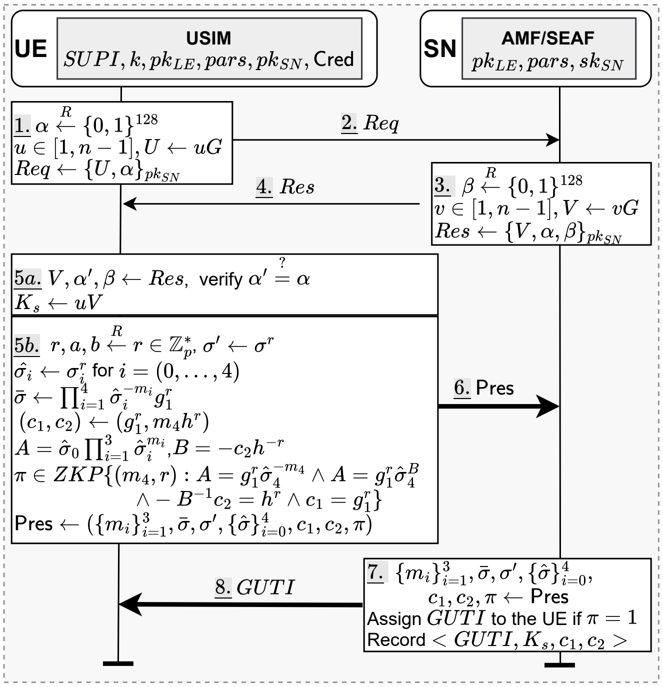
UE
USIM
$SUPI, k, pk_{LE}, pars, pk_{SN}$, Cred
SN
AMF/SEAF
$pk_{LE}, pars, sk_{SN}$
1. $\alpha \overset{R}{\leftarrow} \{0,1\}^{128}$
$u \in [1, n-1], U \leftarrow uG$
$Req \leftarrow \{U, \alpha\}_{pk_{SN}}$
2. Req
3. $\beta \overset{R}{\leftarrow} \{0,1\}^{128}$
$v \in [1, n-1], V \leftarrow vG$
$Res \leftarrow \{V, \alpha, \beta\}_{pk_{SN}}$
4. Res
5a. $V, \alpha', \beta \leftarrow Res$, verify $\alpha' \overset{?}{=} \alpha$
$K_s \leftarrow uV$
5b. $r, a, b \overset{R}{\leftarrow} r \in \mathbb{Z}_p^*$, $\sigma' \leftarrow \sigma^r$
$\hat{\sigma}_i \leftarrow \sigma_i^r$ for $i = (0, \ldots, 4)$
$\bar{\sigma} \leftarrow \prod_{i=1}^4 \hat{\sigma}_i^{-m_i} g_1^r$
$(c_1, c_2) \leftarrow (g_1^r, m_4 h^r)$
$A = \hat{\sigma}_0 \prod_{i=1}^3 \hat{\sigma}_i^{m_i}, B = -c_2 h^{-r}$
$\pi \in ZKP\{(m_4, r) : A = g_1^r \hat{\sigma}_4^{-m_4} \wedge A = g_1^r \hat{\sigma}_4^B$
$\wedge - B^{-1} c_2 = h^r \wedge c_1 = g_1^r\}$
Pres $\leftarrow (\{m_i\}_{i=1}^3, \bar{\sigma}, \sigma', \{\hat{\sigma}\}_{i=0}^4, c_1, c_2, \pi)$
6. Pres
7. $\{m_i\}_{i=1}^3, \bar{\sigma}, \sigma', \{\hat{\sigma}\}_{i=0}^4,$
$c_1, c_2, \pi \leftarrow$ Pres
Assign $GUTI$ to the UE if $\pi = 1$
Record $< GUTI, K_s, c_1, c_2 >$
8. $GUTI$

Fig. 3: The **Presentation and Verification** protocol, which completes the primary Authentication and Key Agreement function between UE and SN. — Per **UE Registration** The thin arrows (i.e., 2, 4) represent the *insecure* channels. The bold arrows (i.e., 6, 8) represent the *secure* channels protected by K_s. Note that, $h = pk_{LE}$.
* An original typo has been corrected in the last line of 5b, Pres. The correct values to be sent to the SN for proving π are $A, -B^{-1}c_2$ instead of A, B. The details of the NIZK computation remain as outlined in the Appendix.E.

can be obtained by a UE through various methods, e.g., HN can pre-install an EF file into USIM containing the frequently used operators' public keys. Throughout the following descriptions, we will specify which computations must be performed by USIM for certain security reasons and which can be optionally executed by ME due to performance considerations.

This phase consists of 5 algorithms: $(\mathsf{Req}, \mathsf{Res}, \mathsf{SNAuth}, \mathsf{PresGen}, \mathsf{Verify})$ and we describe them as follows.

(1, 2) **Connection Request.**
$(\alpha, u, U) \leftarrow \mathsf{Req}(1^\lambda)$
To secure the request during transmission over an insecure channel, in step 1, USIM generates a random nonce α (as a challenge), a temporary EC private key $u \in [1, n-1]$ (used to produce the shared session key later), and the corresponding public key uG. The private key u should only be kept within the secure element, USIM, and thus cannot be abused by ME. USIM then encrypts the two elements (α, uG) with SN's public key pk_{SN} before delivering them as the request Req (step 2). Note that, as the ECDH process also happens during the first stage of SUCI Concealment (i.e., ECIES), we assume USIM and MNOs will still use the standardized domain parameters (e.g., *Curve 25519* for EC key generation as defined in TS 33.501 [11] Annex C 3.4 Profile A) for computing the ECDH keys.

(3, 4) **SN Identification and Session Key Generation.**
$(\alpha, \beta, v, V, K_s) \leftarrow \mathsf{Res}(\mathsf{Req})$
The Req is decrypted under sk_{SN} as U, α. Then SN takes an EC key $v \in [1, n-1]$, and computes the EC public key $V = vG$. Before serving UE, SN needs to prove its authenticity to UE implicitly, by correctly responding to UE with the decrypted challenge α. Besides, SN takes a random challenge nonce β, to prevent potential credential abuse from a malicious ME (discussed in the next step). In step 4, Res will be sent to UE. It comprises the signed hash result of V, α, and β using sk_{SN}, along with the plaintext of these three elements. Besides, SN will compute the shared ECDH key as $K_s = vU$.

It's worth noting that there's no need to protect these three elements. Without knowledge of the temporary EC private key u or v, a passive attacker won't be able to generate a valid shared session key K_s. All further communications occur in a secure channel under K_s.

(5a) **SN Authentication and Session Key Generation.**
$(\alpha', \beta, V, K) \leftarrow \mathsf{SNAuth}(\mathsf{Res}, \alpha, \mathsf{u})$
In step 5a, the hash result over V, α', and β is recomputed and checked against the decrypted hash under pk_{SN}. USIM thus can implicitly authenticate SN's identity by checking if α' is equal to the challenge α that she encrypted by pk_{SN} in step 1. If successful, USIM will calculate the *shared session key* $K_s = uV$ and use it to secure the subsequent communications. The random nonces α and β shall not be revealed to ME, and the β will need to be bound to the one-time verifiable credential Pres later to fulfill UE authentication, which can help to prevent a malicious ME from forging or reusing a valid Pres.

(5b, 6) **Presentation Generation.**
$(\mathsf{Pres}) \leftarrow \mathsf{PresGen}(\sigma, \{\sigma_\mathsf{i}\}_{\mathsf{i}=0}^4, \mathsf{h})$
To fulfill UE registration, a UE is required to present a one-time anonymous presentation (i.e., a one-time verifiable credential), Pres, derived from Cred, to show the validity of its subscriber identity and subscription status. As mentioned before, m_1, m_2, and m_3 are public attributes that will be transmitted in plaintext during each presentation. However, m_4 is a private attribute that must be concealed in Pres to keep the UE's SUPI hidden. The Pres *preparation* process 5b involves **i)** Cred BLINDING and **ii)** IDENTITY ESCROW FUNCTION GENERATION.

i) Cred BLINDING.
To ensure each *UE Registration* is anonymous, as well as different *UE Registration* unlinkable and indistinguishable, the original signatures σ and $\{\sigma_i\}_{i=0}^4$ within the issued credential Cred should be blinded each time but still be able to be authenticated during the presentation. To do so, UE randomly takes a blinding factor $r \in \mathbb{Z}_p^*$, sets $\sigma' = \sigma^r$ and $\hat{\sigma}_i = \sigma_i^r$ for $i = (0, ..., 4)$.

In addition, UE is required to prove that: (1) the randomized signatures $\sigma', \{\hat{\sigma}_i\}_{i=0}^4$ are correctly formed from a set of authentic signatures; (2) The attributes $\{m_i\}_{i=0}^4$ embedded in the credentials are valid, i.e., signed by UE's HN.

Eliminating Pairing Operations on UE side. The original methods [15, 23] for proving possession of such Boneh-Boyen signatures requires the prover to compute at least one pairing operation in addition to a ZKP π. To alleviate the computing burden of the prover in our scheme, i.e., UE, we introduce an auxiliary precomputing parameter $\bar{\sigma} = \prod_{i=1}^{4} \hat{\sigma}_i^{-m_i} g_1^r$ to eliminate pairing operations on the prover side and thus simplify UE's computations during a presentation. (The correctness is discussed later.)

Non-designated Verifiers. The previous BBS-KVAC scheme [22] (see Section IV-D) assumes that the verifier also serves as the issuer, i.e., a verifier has access to the credential issuing/signing keys, allowing the verification computations to be significantly simplified and accelerated. However, we need to accommodate a more general and flexible case in which a verifier is not necessarily the issuer. We extend BBS-KVAC to support *non-designated verifiers*, e.g., verifiers do not necessarily acquire the signing keys. Pairing computation is inevitable under such circumstances to fulfill AC verification. Nevertheless, with our auxiliary parameter $\bar{\sigma}$, the verifier will only need to check if $e(\bar{\sigma}, g_2) \overset{?}{=} e(\sigma', X_0)$, in which two pairings are required. As MNOs are usually considered to have significant computing capability, pairings only incur negligible computing costs to the verifiers.

To summarize, proving possession of Cred requires UE to prove knowledge of the concealed m_4 and the corresponding signatures, without revealing any of them. These will be done on the SN side by verifying a ZKP π', plus a pairing equality check:

$$\begin{aligned} \pi' \in ZKP\{(m_4, r) : \hat{\sigma}_0 \prod_{i=1}^{3} \hat{\sigma}_i^{m_i} = g_1^r \hat{\sigma}_4^{-m_4}\} \\ e(\bar{\sigma}, g_2) \overset{?}{=} e(\sigma', X_0) \end{aligned} \tag{4}$$

Lemma 1. *This construction forms a zero-knowledge proof of knowledge of the signatures σ, $\{\sigma_i\}_{i=0}^{4}$, and* ***witnesses*** *r, m_4 such that $\hat{\sigma}_0 \prod_{i=1}^{3} \hat{\sigma}_i^{m_i} = g_1^r \hat{\sigma}_4^{-m_4}$, with $\sigma \neq 1 \in \mathbb{G}_1$, $\{\sigma_i\}_{i=0}^{4} \neq 1 \in \mathbb{G}_1$. The proofs of* COMPLETENESS, SOUNDNESS *and* ZERO-KNOWLEDGE *are provided in Appendix C.*

ii) IDENTITY ESCROW FUNCTION GENERATION.
To achieve the lawful interception goal, we define an `Identity Escrow` function, i.e., escrow m_4 for future target de-anonymization under the legal circumstance. An escrowed identity should be generated by UE during each presentation, and can only be revealed in the presence of LEA. MNOs cannot deduce any meaningful information related to the UE, or link multiple registrations to a particular UE. First, UE encrypts her m_4 under LEA's public key h via an Elgamal encryption as discussed in Section IV-E, the resulting escrowed identity tuple is $(c_1, c_2) = (g_1^r, m_4 h^r)$.

Commitment over Escrowed Identity. Besides, UE requires to attach a proof to *commit* to the SN that this escrowed identity (c_1, c_2) is genuinely generated from the m_4 contained in the issued Cred, but not a fabricated one. That is, UE should additionally prove that: (1) c_1 is indeed g_1^r, (2) the m_4 appears in the relations $c_2 = m_4 h^r$ and $\hat{\sigma}_0 \prod_{i=1}^{3} \hat{\sigma}_i^{m_i} = g_1^r \hat{\sigma}_4^{-m_4}$ are equal, while the proof π' (equation 4) helps to demonstrate that the m_4 is authentic.

PROOF CONSTRUCTION.
To make the aforementioned proofs about **(i)** and **(ii)** more compact and efficient, we construct a single $\Sigma-$protocol for proving knowledge of all the required *witness* parameters (i.e., attribute m_4 and the blinding factor r) and their relations in one shot.

Two auxiliary parameters A, B are introduced in order to simplify the representations of the statements that we aim to prove in ZKP. Let $A = \hat{\sigma}_0 \prod_{i=1}^{3} \hat{\sigma}_i^{m_i}$, as a result, the ZKP statement to be proven in π' can be rewritten as $A = g_1^r \hat{\sigma}^{-m_4}$. A can be treated as a constant value as all the parameters (i.e., $\{\hat{\sigma}_i\}_{i=0}^{4}$, $\{m_i\}_{i=1}^{3}$) appear in A will be available to SN. Now UE needs to prove that the m_4 within the escrowed identity tuple c_2 is equal to the m_4 in $A = g_1^r \hat{\sigma}_4^{-m_4}$, and this relation can be further represented as $\hat{\sigma}_4^{c_2 h^{-r}} = A^{-1} g_1^r$. Let us denote $B = -c_2 h^{-r}$, which implies that the *commitment* that UE requires to prove is equivalent to the relations:

$$A = g_1^r \hat{\sigma}_4^B, -B^{-1} c_2 = h^r$$

We combine all statements in an AND proof using the general notation (in Eq. (1)), and express the ultimate ZKP π as:

$$\begin{aligned} \pi \in ZKP\{(m_4, r) : A = g_1^r \hat{\sigma}_4^{-m_4} \wedge A = g_1^r \hat{\sigma}_4^B \\ \wedge - B^{-1} c_2 = h^r \wedge c_1 = g_1^r\} \end{aligned} \tag{5}$$

There are four discrete logarithm statements that need to be proven with respect to the secrets m_4 and r. All the parameters other than m_4 and r will be sent to the SN in step 6.

Lemma 2. *Along with $e(\bar{\sigma}, g_2) \overset{?}{=} e(\sigma', X_0)$, this construction forms a zero-knowledge proof of knowledge of σ, $\{\sigma_i\}_{i=0}^{4}$, and* ***witnesses*** *r, m_4 such that $\hat{\sigma}_0 \prod_{i=1}^{3} \hat{\sigma}_i^{m_i} = g_1^r \hat{\sigma}_4^{-m_4}$, with $\sigma \neq 1 \in \mathbb{G}_1$, $\{\sigma_i\}_{i=0}^{4} \neq 1 \in \mathbb{G}_1$, and $c_1 = g_1^r$, $c_2 = m_4 h^r$.*

Our scheme is running under ROM assumptions, and the ZKP process will be run *non-interactively* through a standard Fiat-Shamir heuristic technique. Please refer to Appendix D for detailed proof constructions and computations.

The final challenge $C = H(\bar{y_1} || \bar{y_2} || \bar{y_3} || \beta)$ (produced during running the NIZK process) included in π is a hash digest computed by USIM, towards the concatenation of the commitment values $\bar{y_i}$ (commitments over m_4 and r), and the random challenge β (UE received from SN in step 4) that is used to ensure the Pres is uniquely bound to this session. All the necessary information for a presentation will be represented as Pres by the end of step 5b, and sent to **SN** over a secured communication channel under the session key K_s established between USIM and SN in 5a.

(7, 8) **SN Verification and UE Registration.**
$(C', \bar{\sigma}, \sigma', \{\hat{\sigma}_i\}_{i=0}^{4}, c_1, c_2, \pi, \{\bar{y_i}\}_{i=1}^{4}) \leftarrow \mathsf{Verify}(\mathsf{Pres})$
Step 7 involves *subscription validity checking*, *UE's identity authenticity verification*, and a *double-spending check*. Once received the presentation Pres, SN parses it and performs the following steps:

1) Verify m_1 and m_2 and make an initial assessment of the UE's current subscription validity. Reject the registration request if the credential has expired; otherwise, proceed with further authentication steps.
2) SN identifies the HN to which the UE belongs and retrieves the relevant public verifying parameters X_0 based on m_3.

3) This is a NIZK verification process against the proof π; we only describe the basic operations here, and the Appendix D contains a detailed correctness illustration. Assume $y_1 = A$, $y_2 = A\hat{\sigma}_4^{-B}$, $y_3 = -B^{-1}c_2$, $y_4 = c_1$, and compute: $\bar{y_1} = g_1^{a'}\hat{\sigma}_4^{-b'}y_1^{-C}$, $\bar{y_2} = g_1^{a'}y_2^{-C}$, $\bar{y_3} = h^{a'}y_3^{-C}$, $\bar{y_4} = g_1^{a'}y_4^{-C}$. Then SN recomputes a hash digest C' towards the concatenation of $\bar{y_1}$, $\bar{y_2}$, $\bar{y_3}$ and the random nonce β he generated in step 3, as $C' = H(\bar{y_1}||\bar{y_2}||\bar{y_3}||\beta)$. Continue to the next step if $\bar{y_2} = \bar{y_4}$ and $C = C'$, otherwise, reject the registration request. By far, the UE's commitment to the escrowed identity has been verified; besides, the conformation of β excludes Pres reusing and makes the Pres *non-transferable*.
4) By performing a pairing check $e(\bar{\sigma}, g_2) \overset{?}{=} e(\sigma', X_0)$, SN now conforms UE's possession of a valid credential Cred, i.e., UE has successfully proved that she knows a set of valid BB signatures on the attributes.
5) If the verification succeeds, the corresponding AMF will generate a temporary identifier (i.e., GUTI) and allocate to this UE; besides, the escrowed identity (c_1, c_2), HN identifier m_3, and shared session key K_s, together with GUTI will be recorded as a mapping vector. Finally, the fresh GUTI will be sent to UE over the secure channel protected under K_s, which indicates a successful UE registration.

D. UE Deregistration

As mentioned in Section II, GUTI can be used by UE while within the same AMF region (i.e., one TA or multiple TAs) that she registered with, presenting a valid GUTI can eliminate a full AKA process. The temporary session key K_s is comparable to the K_{SEAF} established in 5G-AKA, which can be used to secure the subsequent communications between UE and the SN. Current 5G systems improve inter-AMF mobility and allow for the use of a single GUTI in multiple neighbor TAs to reduce the frequency of UE registration when a UE moves between TA/RAs that are under different AMF regions. However, in order to minimize the UE footprint as much as possible while attaching to different cells (i.e., gNBs) and avoiding linkability risks (discussed in [21, 48]) even to a single pseudonym (i.e., GUTI), our scheme makes a trade-off between UE privacy and granularity of AMF binding, and requires that UE de-register herself once moving outside of the AMF that she registered with, and initiate a new registration while passing the TA border. SN (or UDM) thus notifies the old AMF and de-activates the UE context along with the old GUTI. The context shall be stored in SN's database, and the storage time is dictated by LI requirements.

E. Lawful De-anonymization

Lawful de-anonymization requires collaboration from both SN, HN, and an authorized LEA (in non-roaming cases, SN = HN). Although LI-related NFs are deployed within the core networks and controlled by the honest-but-curious MNOs, our scheme ensures that any MNO can't de-anonymize certain UE independently without a LI process that is initialized by LEA.

As per the LI architecture defined in TS 33.107 and TS 33.126 [9, 10], Law Enforcement Monitoring Facility (LEMF) is the only component that resides in the LEA domain, and the LI products (e.g., target UE's identity) will be eventually provided to the LEMF.

This method can be used to identify target UEs within specific areas during a time frame, such as identifying riot protesters or investigating suspects involved in a bank robbery, under a "Geofence Search Warrant". After receiving the related information, SN should identify the GUTIs that meet such conditions and send the correlated escrowed identity (c_1, c_2) to LEA. We assume that only with the presence of a TPM (e.g., a hardware token or a specialized device that stores LEA secret keys) possessed by LEA can operate with LEMF and assist in decrypting the *Identity Escrow* components. LEA with authorized TPM devices can thus recover the m_4, i.e., $MSIN$, by computing $s = c_1^{sk_{LE}}$ and $m_4 = c_2 s^{-1}$.

SUPI can eventually be obtained by combining the decrypted m_4 with m_3 (i.e., MCC, MNC), which UE presented to SN during the registration phase. Then the target SUPI will be input to the following LI NFs (e.g., LEMF, MDF, LIPF and LICF) residing in HN to produce further meaningful information requested by LEA, such as the real-world identity of the target UE's owner and home address that subscriber provided to the carrier during the subscription.

AAKA purposefully makes it more difficult to conduct mass surveillance, by introducing the *Identity Escrow* function to perform lawful de-anonymization. It is worth noting that, AAKA not only provides the abovementioned Geofence-based de-anonymization but also supports searching for a known target (i.e., locate the target given a known SUPI) by incorporating a dedicated TPM (e.g., TEE) to the LI NFs. This integration allows for real-time decryption of escrowed identities. If a match is found between the escrowed identity and the target SUPI, the TPM will output the target's metadata, enabling further LI actions such as monitoring communication content.

VI. SECURITY ANALYSIS

In this section, we will analyze the security and privacy properties of AAKA. Due to space limitations, some proofs of the theorems will be covered in the Appendices.

A. Security and Privacy Properties

Theorem 2 (AAKA SECURITY AND CORRECTNESS)**.** *If DDH and SCDHI (see Appendix B) problems hold and the zero-knowledge proof system satisfies the properties outlined in Lemma 1, the algorithms (*Setup, KeyGen, Issue, Obtain, PresGen, Verify*) collectively form a secure keyed-verification credential scheme.*

The unforgeability of the credential is derived intuitively from the unforgeability of the BBS-based MAC (Theorem 1). It ensures that an adversary, denoted as $\mathcal{A}$, cannot forge a legitimate $Cred$ or a valid proof of a valid attribute set that is signed by the HN's issuing key ik:

Theorem 3 (UNFORGEABILITY)**.** *The* PresGen, Verify *in the Presentation Generation and Verification protocol for* KeyGen, Issue *is (t, Q, ϵ)-unforgeable if for all PPT adversaries $\mathcal{A}$ that make at most Q random oracle queries and run in time t:*

$$
\begin{aligned}
&Pr[(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, p) \xleftarrow{\$} \mathsf{Setup}(1^\lambda),\\
&\quad (par, ik, h, sk_{LE}) \xleftarrow{\$} \mathsf{KeyGen}(\mathsf{pars}),\\
&\quad (\mathsf{state}, \phi) \leftarrow \mathcal{A}(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, p, par)^{\mathcal{O}_I(ik,\cdot), \mathcal{O}_V(\cdot,\cdot)}\\
&\quad \mathcal{A}(\mathsf{state} \leftrightarrow \mathsf{Verify}(\phi)) \leftarrow b :\\
&\quad b = 1 \wedge (\forall (m_1, ..., m_4) \in Q,\\
&\quad \phi(m_1, ..., m_4) = 0)] \leqslant negl(\lambda)
\end{aligned}
$$

Where $\mathcal{O}_I(ik, \cdot)$ denotes the issue oracle, and $\mathcal{O}_V(\cdot, \cdot)$ represents the presentation verification oracle. ϕ corresponds to the set of attributes supported in this credential scheme, while Q denotes the collection of all attributes transmitted to the $\mathcal{O}_I(ik, \cdot)$ oracle. Let $\mathcal{M}$ denotes the attribute set $(m_1, ..., m_n)$ ($(n = 4)$ in AAKA), in which $m_n \in \mathbb{Z}_p^*$.

Theorem 4 (Anonymity and Unlinkability across different UE registrations)**.** *AAKA offers both unlinkability and anonymity if for all PPT adversaries $\mathcal{A}$, there exists an efficient simulator* SimPres *such that for all the attributes $(m_1, ..., m_4) \in \mathcal{M}$ such that $\phi(m_1, ..., m_4) = 1$, and for all $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, p) \xleftarrow{\$} \mathsf{Setup}(1^\lambda)$, $(par, ik, h, sk_{LE}) \xleftarrow{\$}$* KeyGen(pars)*, for all $Pres$ such that* Verify(Pres) *output 1:*

$$
\mathsf{Show}(par, \mathsf{Pres}, (m_1, ..., m_4), \phi) \leftrightarrow \mathcal{A} \rightarrow \mathsf{state} \approx \mathsf{SimPres}(par, ik, \phi)
$$

i.e., the adversary $\mathcal{A}$'s view given the proof can be simulated by SimPres *given only ϕ and a valid credential issuing key ik corresponding to the public verification parameters par. According to Chase et al. in [32], if anonymity in an AC scheme is maintained across multiple presentations of the credential, it is considered to meet the criteria of multi-show* unlinkability*, and these two properties can be proved by the same game.*

In a nutshell, our one-time verifiable credential Pres enables anonymous registration of a UE with a specific MNO, ensuring that the subscriber's identity (i.e., m_4) remains undisclosed to any MNO. Moreover, multiple Pres derived from the same Cred are unlinkable, while different Pres derived from different UE's Cred are indistinguishable. These properties are guaranteed by the zero-knowledge property of the proofs (discussed in Appendix C) under the random oracle model. In essence, the proof π produced in step 5b for Pres appears random and it reveals only the validity of the 4 attributes $\{m_i\}_{i=1}^4$ (with m_4 being the only hidden attribute). As $\{m_i\}_{i=1}^3$ only indicates UE's HN ID and date of expiration (e.g., 060123), numerous UEs from the same HN could possess the same $\{m_i\}_{i=1}^3$ in their Pres, making them hard to be distinguished, thus, revealing $\{m_i\}_{i=1}^3$ won't affect the Anonymity and Unlinkability properties of our scheme. Thereby, AAKA ensures different registrations made by either the same UE or different UEs cannot be correlated or identified, rendering it impossible for any MNO alone to trace a specific UE.

B. *Outsider Attack*

As **Credential Issuance** phase is protected by the existing OTA provisioning process, we consider this sub-protocol inherently resistant to outsider attacks, i.e., the Cred can be securely stored into SIM, and cannot be extracted by any party without SIM administrative keys of HN.

The outsider can interfere with the insecure channel used by the **Presentation and Verification** sub-protocol (Fig.3, steps 2 and 4) before completion of the key agreement. Nevertheless, this sub-protocol is resistant to *replay attack*. This is primarily due to the fact that UE's temporary private key u is always safeguarded within the SIM, where even a semi-trusted ME can't acquire it. Even if an eavesdropper intercepts the temporary SN public key V, and the random nonces α, β transmitted during step 2, they cannot produce the valid shared key K_s required for establishing the secure channel with SN. Besides, any tampering of V, α, and β can be detected by checking the signed hash value in step 5a.

Secure Computation Splitting and Non-transferability. Most removable cellular SIMs have limited computing capabilities, including limited memory size, outdated cryptographic co-processors, and lack of support for Java Card 3.0.x crypto API, etc. This is because subscribers often reuse their old SIMs obtained before the 5G era. Consequently, according to TS 33.501 [7, 11], the ME in current 5G systems can optionally assist with complex cryptographic operations (e.g., SUCI concealment), and key derivation and storage (e.g., K_{AUSF}, K_{AMF}, K_{SEAF}, etc.).

AAKA aims to strike a balance between security and efficiency. To minimize computing overhead, we restrict critical operations to the SIM while delegating resource-intensive operations to the *semi-trusted* ME, ensuring protocol security remains unaffected.

Specifically, step 1 and 5a are exclusively executed by the SIM, ensuring that the key u and K_s remain within the SIM and are never exposed. However, the computation of 5b can be performed by the ME. Recall that a valid Pres in step 5b requires a correct β that matches SN's challenge, a dishonest ME cannot fabricate a valid Pres without knowledge of the one-time value β. Thus, the *non-transferability* is ensured.

VII. Evaluation

A. *Experimental Environment*

The current standard cellular SIM cards are all based on the Java Card environment for USIM application development, in which a dedicated cryptographic coprocessor and corresponding crypto APIs (e.g., `javacard.security`, `javacardx.crypto` [54]) are available. USIM applications (Java Card Applets) can be OTA provisioned into USIM by operators with administrative keys that are only known by HN. Our proof-of-concept prototype includes:

1) **MNO**, implemented on a standard PC (Intel Core i7-11700k, 3.6GHZ, 8-core, 64-bit CPU with a Linux OS);
2) **UE**, a constrained host device (i.e., ME) that is equipped with a standard cellular SIM card (**Card A**, sysmoISIM-SJA2, 64KB EEPROM), which supports OTA provisioning and contains off-the-shelf legacy programmable cellular file structures (EF files, e.g., EF_{IMSI} for IMSI-Type SUPI, EF_{ORPK} stands for pk_{HN}), but it only supports up to Java Card SDK 2.2.1 (unfortunately, to the best of our knowledge, it is the only programmable standard 5G cellular SIM card.); and a general-purpose Java Card (**Card B**, NXP JCOP J3R110), which

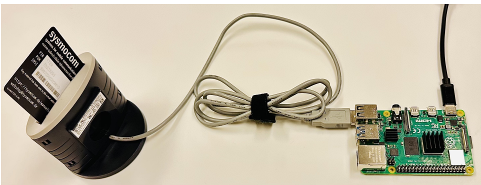

Fig. 4: Testbed of ME

supports the up-to-date Java Card SDK 3.0.5 and allows for implementing certain complex cryptographic operations (e.g., EC point scalar, SHA–256).

As mentioned in Section III-B1, modifying a standard cell (e)SIM without carrier privilege causes mobile network connection failure due to proprietary and security restrictions. Given the diversity of cellular-connected devices, we emulate the **ME** by a Raspberry Pi 4, for which both the processor (1.5GHz 64bit quad-core Cortex A72 ARM v8) and memory size (4GB RAM and 64GB SD card) are comparable to an oldish mid-range Android phone[4]. Two SIM cards with different models are used to measure the running cost of our scheme. As shown in Fig. 4, we use a PC/SC compliant smart card reader (Omnikey) to connect the SIM cards (**Card A, B**) with the ME and transfer APDU commands towards cards with hardware-dependent administrative keys for R/W operations.

B. Implementation and Parameters

Our implementation adopts standard cryptographic parameters (*Curve25519* in Montgomery form, and ANSI-X9.63 KDF) that are suggested in 3GPP 33.501 [11] (Annex C 3.4 Profile A) for SUCI (de)concealment computations involved in both 5G-AKA and our AAKA (during Cred *Issuance* process). The symmetric encryption algorithm in ECIES is AES-128 in CTR mode. All the hash digests involved in our implementation are generated from SHA-256.

The curve and pairings used in AAKA Cred, Pres are the efficient pairing-friendly BN (Barreto-Naehrig) curve [13], `BN-254` (also known as `alt_bn_128`, or `BN-256`), with approximately 100-bit security level due to recent attacks [12]. The prime modulus p is well-defined as $p(x) = 36x^4 + 36x^3 + 24x^2 + 6x + 1$ with an embedding degree of 12, i.e., the group $\mathbb{G}_1$, $\mathbb{G}_2$ are defined over $\mathbb{F}_p$, $\mathbb{F}_{p^2}$, respectively, where the target group is defined over $\mathbb{F}_{p^{12}}$. The `BN-254` testing is implemented via `mcl`, a pairing-based cryptography C++ library [1].

On ME, C- and Python-based communication APIs (e.g., `pyscard`, `pcsc`) were used to manage communications between ME and SIM. We use `pySim` to read (binary decoding) and modify the standard cellular EF files, e.g., obtain the legacy MNC-MCC, SUPI, and pK_{HN}; and a series of Java Card development toolkits (e.g., `GlobalPlatform Pro`) to manage Java Card Applets on USIM.

All the computation times mentioned below are the average of 10 runs in milliseconds, results are shown in Table III.

TABLE III: **Time Consumption** (in milliseconds) of different stages in AAKA Protocol

HN		UE			SN	
Issue	Verify	Obtain	Req + SNAuth	PresGen	Res	Verify
1.87	1.09	38.66	131.30	51.72	0.078	4.51

A monthly *credential issuance* (Issuce) computation on PC (i.e., HN) takes 1.87 ms, while an Obtain on ME takes 38.66ms, the Cred will be stored on USIM as a binary EF file after a successful issuance.

During *Presentation Generation and Verification*, AAKA confines the EC private key generation (u in Req, step 1 of Fig. 3) and ECDH shared key computation (K_s in SNAuth, step 3) within SIM. The EC key pair generation and ECDH key computation are implemented as a Java Applet (a *.cap* file loaded into **Card B**) via on-card cipher suites. The on-card generation of the 256 bits private key (one ECC operation over `Curve 25519`) takes 98.11ms, while the computation of the shared key (i.e., the agreed session key K_s) takes 33.19ms. It takes 0.078 ms for SN (i.e., PC) to compute a Res.

ME, as the computing helper device, performs credential blinding and identity escrow computation. It takes 51.72ms to generate a Pres (i.e., 5b PresGen algorithm) on the UE side.

The credential verification procedure in *roaming* mode (e.g., run Verify algorithm by SN) takes less than 4.51ms on the PC, in which SN needs to compute 2 pairing operations and 6 exponentiation in g_1 (discussed in Appendix D). We also measure the *non-roaming* mode, i.e., the verifier is the issuer (HN), via the original methods proposed by Camenisch et al. [22], and it takes 1.09ms for a successful verification as HN obtains the issuing key, thereby eliminating the expensive pairing equality check.

C. Comparison with the Current 5G Solution

To demonstrate the feasibility of our solution, we compare our time consumption with the current UE registration process by implementing a 5G-AKA protocol within the same hardware environments as the benchmark. To eliminate unnecessary impact factors, we do not consider the transmission latency introduced from **a.** communications between different NFs within a core network (e.g., AMF, SEAF, UDM, AUSF); and **b.** communications between the core network, ngRAN, and UE. As per TS 31.102 Section 5.3.47 [8], 3GPP defines an alternative option for executing the authentication process when SIM is incapable of executing ECIES: if the *Service 124*, $EF_{SUCI\_CALC\_INFO}$ in USIM is enabled (indicated by HN beforehand), SUCI concealment shall be calculated by ME. Thereby, on the UE side, we compare the results in both the hybrid setting and ME-only setting:

Test choice (1): Hybrid USIM + ME. In 5G-AKA, the computation on the UE side involves a SUCI Concealment (i.e., ECIES, step 1 in Fig. 5, Appendix A), lightweight XORs and MAC through the prefixed on-card KDFs f_1, f_2, f_5 set by MNOs, and a sequence number SQN comparison (step 7). Among them, the major computation cost is brought by the ECC operations in the ECIES. In general, ECIES consists of four stages: ECDH + x9.63 KDF + AES-128 CTR + SHA256-HMAC. Due to the lack of on-card support for the X9.63

[4] For example, Qualcomm's Snapdragon 620/650/652 series (Android-based) mobile device SoC that came out around 2015 are equipped with quad-core Cortex-A72 CPU.

TABLE IV: **Time Consumption** (in milliseconds) on different entities under AAKA and 5G-AKA

	ME only	ME+USIM	SN	HN
5G-AKA	3.69	124.04 + τ	3.81 + Δ*	
AAKA	52.92	184.22	4.59	1.07

*During roaming, authentication is collaboratively executed by SN and HN in 5G-AKA.

KDF, we are not able to precisely deploy a full ECIES process executed by SIM, but only measure the on-card ECDH stage as 124.04 ms (on **Card B**). Nevertheless, as ECDH is considered to be far more costly than the rest of the operations, we denote the estimated running time of the resting steps as τ. Thus, the total computations (1+7) on UE during 5G-AKA is denoted as (124.04 + τ) ms. As discussed previously, the total execution time (Req + SNAuth + PresGen) on the hybrid *USIM + ME* setting under AAKA is 184.22 ms.

Test choice (2): ME Only. Under this setting, we implement the computations involved on the UE side of the 5G-AKA and AAKA process purely in ME, while the connected **Card A** only provides legacy EF files to ME. The standard cryptography library `pyca` [2] (`x25519`, KDF `x963kdf`, SHA256 `hmac`) is used.

Without the bottleneck caused by slow on-card EC operations, the total time consumption during 5G-AKA on the UE side drops to 3.69 ms. For AAKA, the running time on the UE side decreases from 184.22 ms to 52.92 ms, as the private key generation and ECDH key computation (in Req and SNAuth) only take approx. 1.2 ms on the Raspberry Pi.

On MNO side. For 5G-AKA, the computations on the MNO side are primarily carried out by HN, with SN serving as an intermediary. We measure the total time consumption on the MNO (SN + HN) side as 3.81 ms. In a *roaming* case, there are 4 interactions (i.e., 3, 5, 10, 12 in the 5G-AKA protocol (Fig. 5) between HN and SN, while AAKA does not require such communications between HN and SN. As communication channel conditions between HN and SN usually vary and could bring non-trivial latency compared to 3.81 ms in reality, we use Δ to denote it as an unknown factor. Thus the total time consumption on the MNO side is (3.81 + Δ) ms, and the Δ could be negligible under a *non-roaming* case. In our previous test of AAKA, the verification under *roaming* (on SN) takes 4.59 ms in total, whereas it takes 1.07 ms under a *non-roaming* case (on HN).

Comparisons are summarized in Table IV. When the ME is designated as the helper device, on the UE side, AAKA only adds up to 50% ($\sim$60 ms) computation overhead, compared to the 5G-AKA non-roaming case; and with even less overhead on a roaming case. This is because our solution eliminates the communications between SN and HN during roaming, which compensates the computation cost brought by relatively expensive cryptographic operations in AAKA. We can conclude that, on a constrained host device, AAKA has acceptable execution overhead.

VIII. Conclusions and Future Works

This paper addresses the mobile tracking problem which is rooted in the misuse of subscribers' location information databases collected by MNOs. We consider the most challenging threat model where MNOs are assumed to be the honest-but-curious privacy attackers. Leveraging novel anonymous credential techniques, we proposed AAKA, an anti-tracking mobile access solution that encompasses meticulously designed secure and efficient anonymous credential schemes and authentication protocols. Those components are seamlessly integrated to achieve anonymity, unlinkability, and additionally offer the capability of lawful deanonymization when the LEA is collaborating with the MNOs. Our design is compatible with the current 5G cellular architecture, requiring minimal modifications in the selection of cryptographic algorithms, and our experimental implementations demonstrated the efficacy and efficiency of our scheme, as they are efficient to operate using the readily available mobile hardware.

This paper presents the overall architecture of AAKA and its key components. From the perspective of practical deployment, AAKA inherently supports an unlimited service plan where users pay a monthly fee, obtain a credential with a new expiration date, and enjoy unlimited network access. AAKA can be extended to support a usage-based mobile service plan, where cryptocurrency-like tokens can be generated and distributed to subscribers and those anonymous tokens will be consumed as users access the network. It is important to note that the exploration of this usage-based mobile service plan constitutes a related yet distinct endeavor, which we intend to address in a separate publication.

As a final remark, AAKA addresses mobile user anonymity and the associated location privacy concerns. Once mobile users establish network connectivity, they can leverage a wide range of applications, some may involve end-to-end encryption. Looking ahead, we envision that a hybrid solution with an out-of-band callee discovery and call-routing mechanism can also be achieved by embedding extra options into our credentials, for example, by leveraging the decentralized identifier (DID) framework [63] and VoIP phone applications as in [37], and this is a promising topic for our future work. There may be de-anonymization attacks at higher levels, which must be addressed separately by utilizing techniques such as proxies, Tor, and so on.

Acknowledgment

This work was supported in part by the US National Science Foundation under grants 2247560, 2247561, 2247562, 2154929, and 1916902, the Office of Naval Research under grant N00014-19-1-2621, DARPA under contract HR001120C0155, and a gift from InterDigital.

References

[1] GitHub - herumi/mcl: a portable and fast pairing-based cryptography library — github.com. https://github.com/herumi/mcl.

[2] Welcome to pyca/cryptography x2014; Cryptography 40.0.0.dev1 documentation — cryptography.io. https://cryptography.io/en/latest/.

[3] Big Telecom Sold Highly Sensitive Customer GPS Data Typically Used for 911 Calls. https://www.vice.com/en/article/a3b3dg/big-telecom-sold-customer-gps-data-911-calls, 2019.

[4] I gave a bounty hunter $300. then he located our phone. https://www.vice.com/en/article/nepxbz/i-gave-a-bounty-hunter-300-dollars-located-phone-microbilt-zumigo-tmobile, 2019.

[5] 3GPP TR 33.935 V17.0.0. *Study on detailed Long Term Key Update Process (LTKUP)*, 2022.

[6] 3GPP TS 23.003 V17.5.0. *Numbering, addressing and identification*, 2022.

[7] 3GPP TS 31.102 V16.4.0. *Characteristics of the Universal Subscriber Identity Module (USIM) application*, 2020.

[8] 3GPP TS 33.102 v16.4.0. *Characteristics of the Universal Subscriber Identity Module (USIM) application*, 2020.

[9] 3GPP TS 33.107 V17.0.0. *Lawful Interception (LI) architecture and functions*, 2022.

[10] 3GPP TS 33.126 V16.2.0. *Lawful Interception requirements*, 2020.

[11] 3GPP TS 33.501 v16.9.0. *Security architecture and procedures for 5G System*, 2022.

[12] Razvan Barbulescu and Sylvain Duquesne. Updating key size estimations for pairings. *Journal of cryptology*, 32:1298–1336, 2019.

[13] Paulo SLM Barreto and Michael Naehrig. Pairing-friendly elliptic curves of prime order. In *Selected Areas in Cryptography: 12th International Workshop, SAC 2005, Kingston, ON, Canada, August 11-12, 2005, Revised Selected Papers 12*, pages 319–331. Springer, 2006.

[14] David Basin, Jannik Dreier, Lucca Hirschi, Saša Radomirovic, Ralf Sasse, and Vincent Stettler. A formal analysis of 5g authentication. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 1383–1396, 2018.

[15] Dan Boneh and Xavier Boyen. Short signatures without random oracles and the sdh assumption in bilinear groups. *Journal of cryptology*, 21(2):149–177, 2008.

[16] Andre Bourdoux, Andre Noll Barreto, Barend van Liempd, Carlos de Lima, Davide Dardari, Didier Belot, Elana-Simona Lohan, Gonzalo Seco-Granados, Hadi Sarieddeen, Henk Wymeersch, et al. 6g white paper on localization and sensing. *arXiv preprint arXiv:2006.01779*, 2020.

[17] Stefan Brands. *Rethinking public key infrastructures and digital certificates: building in privacy*. Mit Press, 2000.

[18] Ernie Brickell, Jan Camenisch, and Liqun Chen. Direct anonymous attestation. In *Proceedings of the 11th ACM conference on Computer and communications security*, pages 132–145, 2004.

[19] Ernie Brickell, Liqun Chen, and Jiangtao Li. Simplified security notions of direct anonymous attestation and a concrete scheme from pairings. *International journal of information security*, 8(5):315–330, 2009.

[20] Ernie Brickell and Jiangtao Li. Enhanced privacy id: A direct anonymous attestation scheme with enhanced revocation capabilities. In *Proceedings of the 2007 ACM workshop on Privacy in electronic society*, pages 21–30, 2007.

[21] Ksenia Budykho, Ioana Cristina Boureanu, Stephan Wesemeyer, Daniel Romero, Matt Lewis, Yogaratnam Rahulan, and Fortunat Rajaona. Fine-grained trackability in protocol executions. In *Network and Distributed System Security (NDSS) Symposium 2023*.

[22] Jan Camenisch, Manu Drijvers, Petr Dzurenda, and Jan Hajny. Fast keyed-verification anonymous credentials on standard smart cards. In *IFIP International Conference on ICT Systems Security and Privacy Protection*, pages 286–298. Springer, 2019.

[23] Jan Camenisch, Manu Drijvers, and Jan Hajny. Scalable revocation scheme for anonymous credentials based on n-times unlinkable proofs. In *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society*, pages 123–133, 2016.

[24] Jan Camenisch, Manu Drijvers, and Anja Lehmann. Anonymous attestation using the strong diffie hellman assumption revisited. In *International Conference on Trust and Trustworthy Computing*, pages 1–20. Springer, 2016.

[25] Jan Camenisch and Thomas Groß. Efficient attributes for anonymous credentials. In *Proceedings of the 15th ACM conference on Computer and communications security*, pages 345–356, 2008.

[26] Jan Camenisch, Susan Hohenberger, Markulf Kohlweiss, Anna Lysyanskaya, and Mira Meyerovich. How to win the clonewars: efficient periodic n-times anonymous authentication. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 201–210, 2006.

[27] Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *Annual international cryptology conference*, pages 56–72. Springer, 2004.

[28] Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups. In *Annual International Cryptology Conference*, pages 410–424. Springer, 1997.

[29] Jan Camenisch and Els Van Herreweghen. Design and implementation of the idemix anonymous credential system. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 21–30, 2002.

[30] Carly Page. New documents reveal 'huge' scale of us government's cell phone location data tracking. https://techcrunch.com/2022/07/18/homeland-security-cell-phone-tracking/, 2022.

[31] Melissa Chase and Anna Lysyanskaya. On signatures of knowledge. In *Annual International Cryptology Conference*, pages 78–96. Springer, 2006.

[32] Melissa Chase, Sarah Meiklejohn, and Greg Zaverucha. Algebraic macs and keyed-verification anonymous credentials. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 1205–1216, 2014.

[33] Ivan Damgård. Efficient concurrent zero-knowledge in the auxiliary string model. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 418–430. Springer, 2000.

[34] José A del Peral-Rosado, Jani Saloranta, Giuseppe Destino, José A López-Salcedo, and Gonzalo Seco-Granados. Methodology for simulating 5G and gnss high-accuracy positioning. *Sensors*, 18(10):3220, 2018.

[35] Yvo Desmedt and Yair Frankel. Threshold cryptosystems. In *Conference on the Theory and Application of Cryptology*, pages 307–315. Springer, 1989.

[36] Yevgeniy Dodis, Eike Kiltz, Krzysztof Pietrzak, and

Daniel Wichs. Message authentication, revisited. In *Advances in Cryptology–EUROCRYPT 2012: 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings 31*, pages 355–374. Springer, 2012.

[37] Changlai Du, Hexuan Yu, Yang Xiao, Y Thomas Hou, Angelos D Keromytis, and Wenjing Lou. {UCBlocker}: Unwanted call blocking using anonymous authentication. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 445–462, 2023.

[38] Changlai Du, Hexuan Yu, Yang Xiao, Wenjing Lou, Chonggang Wang, Robert Gazda, and Y Thomas Hou. Mobile tracking in 5g and beyond networks: Problems, challenges, and new directions. In *2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS)*, pages 426–434. IEEE, 2022.

[39] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on information theory*, 31(4):469–472, 1985.

[40] FCC. FCC proposes $12.2m fine against Sprint in location information case. https://www.fcc.gov/document/fcc-proposes-122m-fine-against-sprint-location-information-case, 2020.

[41] FCC. FCC proposes $48.3m fine against Verizon in location information case. https://www.fcc.gov/document/fcc-proposes-483m-fine-against-verizon-location-information-case, 2020.

[42] FCC. FCC proposes $57.2m fine against AT&T in location information case. https://www.fcc.gov/document/fcc-proposes-572m-fine-against-att-location-information-case, 2020.

[43] FCC. FCC proposes $91.6m fine against T-Mobile in location information case. https://www.fcc.gov/document/fcc-proposes-916m-fine-against-t-mobile-location-information-case, 2020.

[44] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. *Journal of cryptology*, 1(2):77–94, 1988.

[45] Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Conference on the theory and application of cryptographic techniques*, pages 186–194. Springer, 1986.

[46] Brian Fung. Wireless carriers keep your location data for years and provide it to the police. https://www.cnn.com/2022/08/29/tech/wireless-carriers-locations-fcc/index.html, 2022.

[47] Steven D Galbraith, Kenneth G Paterson, and Nigel P Smart. Pairings for cryptographers. *Discrete Applied Mathematics*, 156(16):3113–3121, 2008.

[48] Byeongdo Hong, Sangwook Bae, and Yongdae Kim. Guti reallocation demystified: Cellular location tracking with changing temporary identifier. In *NDSS*, 2018.

[49] Ojas Kanhere and Theodore S Rappaport. Position location for futuristic cellular communications: 5G and beyond. *IEEE communications magazine*, 59(1):70–75, 2021.

[50] Bedran Karakoc, Nils Fürste, David Rupprecht, and Katharina Kohls. Never let me down again: Bidding-down attacks and mitigations in 5g and 4g. 2023.

[51] Adrien Koutsos. The 5g-aka authentication protocol privacy. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 464–479. IEEE, 2019.

[52] Ben Lovejoy. Carrier location data usage again under investigation, after promises broken. https://9to5mac.com/2022/07/22/carrier-location-data/, 2022.

[53] Supreme Court of the United States. 16-402 carpenter v. https://www.supremecourt.gov/opinions/17pdf/16-402_h315.pdf, 2017.

[54] Oracle. Java Card Platform, Classic Edition 3.0.5 — docs.oracle.com. https://docs.oracle.com/javacard/3.0.5/index.html.

[55] Oros42/IMSI-Catcher. GitHub - Oros42/IMSI-catcher: This program show you IMSI numbers of cellphones around you. — github.com. https://github.com/Oros42/IMSI-catcher.

[56] Ivan Palamà, Francesco Gringoli, Giuseppe Bianchi, and Nicola Blefari-Melazzi. Imsi catchers in the wild: A real world 4g/5g assessment. *Computer Networks*, 194:108137, 2021.

[57] Christian Paquin and Greg Zaverucha. U-prove cryptographic specification v1. 1. *Technical Report, Microsoft Corporation*, 2011.

[58] Mohit Rathi. Rethinking reverse location search warrants. *The Journal of Criminal Law and Criminology (1973-)*, 111(3):805–837, 2021.

[59] Paul Schmitt and Barath Raghavan. Pretty good phone privacy. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 1737–1754, 2021.

[60] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *Conference on the Theory and Application of Cryptology*, pages 239–252. Springer, 1989.

[61] Yannick Seurin. On the exact security of schnorr-type signatures in the random oracle model. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 554–571. Springer, 2012.

[62] Keen Sung, Brian Levine, and Mariya Zheleva. Protecting location privacy from untrusted wireless service providers. In *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 266–277, 2020.

[63] W3C. Decentralized identifiers (dids) v1.0. https://www.w3.org/TR/2022/REC-did-core-20220719/, 2022.

[64] Yuchen Wang, Zhenfeng Zhang, and Yongquan Xie. Privacy-Preserving and Standard-Compatible AKA Protocol for 5G. In *USENIX Security Symposium*, pages 3595–3612, 2021.

Appendix

A. 5G-AKA

A detailed 5G-AKA protocol is provided in Figure 5. f_1, f_2, f_5 are standardized KDFs that are pre-fixed within cellular SIM cards. $HRES^*$ is the challenge generated by HN, while $HRES$ is the response computed by UE. The comparison between the sequence number SQN_{UE} and SQN_{HN} are designed for message freshness checking and replay protection, although several works showed that it incurs linkability risks. In essence, 5G-AKA is implicitly accomplished through key confirmation round trips.

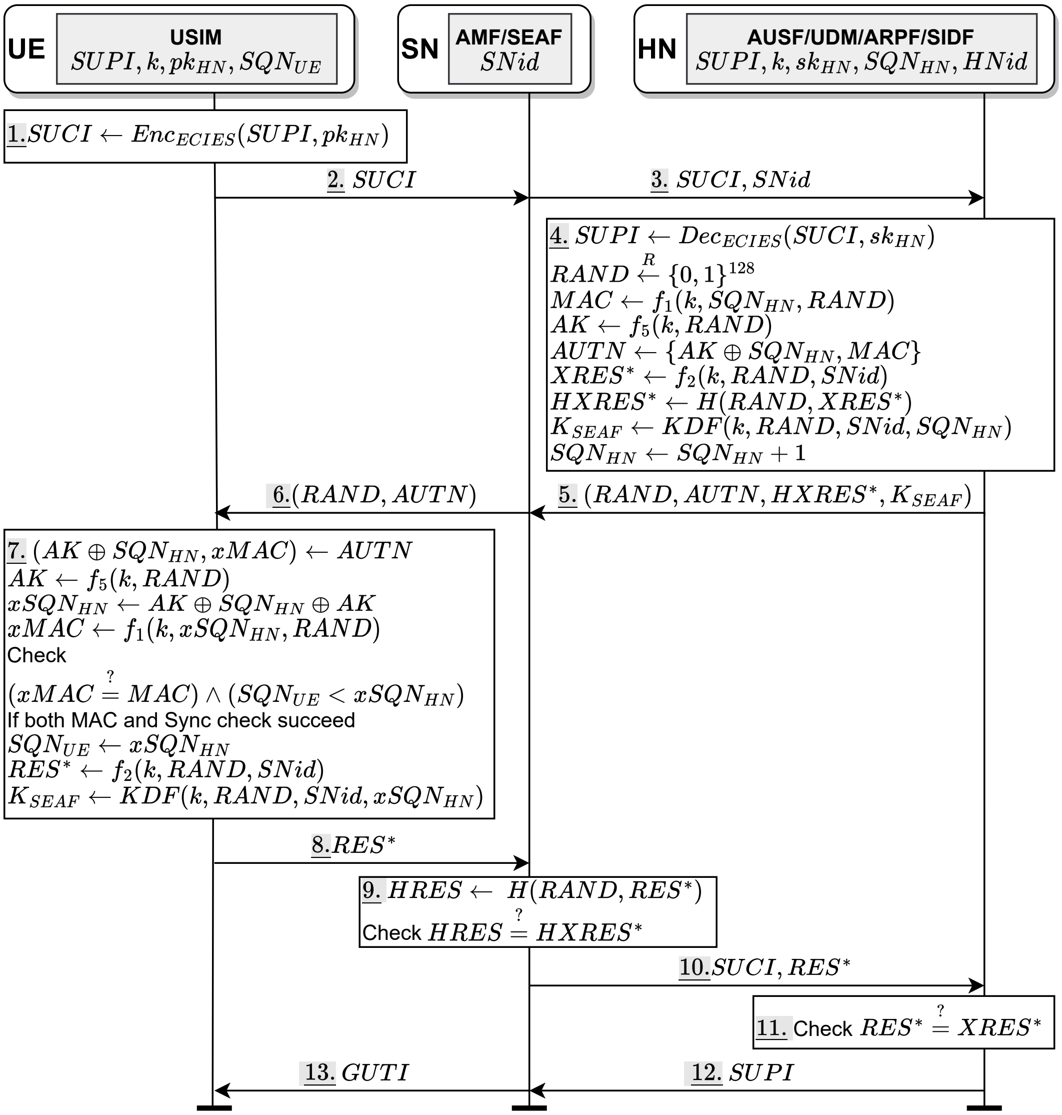
UE
USIM
$SUPI, k, pk_{HN}, SQN_{UE}$
SN
AMF/SEAF
$SNid$
HN
AUSF/UDM/ARPF/SIDF
$SUPI, k, sk_{HN}, SQN_{HN}, HNid$
1. $SUCI \leftarrow Enc_{ECIES}(SUPI, pk_{HN})$
2. $SUCI$
3. $SUCI, SNid$
4. $SUPI \leftarrow Dec_{ECIES}(SUCI, sk_{HN})$
$RAND \overset{R}{\leftarrow} \{0,1\}^{128}$
$MAC \leftarrow f_1(k, SQN_{HN}, RAND)$
$AK \leftarrow f_5(k, RAND)$
$AUTN \leftarrow \{AK \oplus SQN_{HN}, MAC\}$
$XRES^* \leftarrow f_2(k, RAND, SNid)$
$HXRES^* \leftarrow H(RAND, XRES^*)$
$K_{SEAF} \leftarrow KDF(k, RAND, SNid, SQN_{HN})$
$SQN_{HN} \leftarrow SQN_{HN} + 1$
6.$(RAND, AUTN)$
5. $(RAND, AUTN, HXRES^*, K_{SEAF})$
7. $(AK \oplus SQN_{HN}, xMAC) \leftarrow AUTN$
$AK \leftarrow f_5(k, RAND)$
$xSQN_{HN} \leftarrow AK \oplus SQN_{HN} \oplus AK$
$xMAC \leftarrow f_1(k, xSQN_{HN}, RAND)$
Check
$(xMAC \overset{?}{=} MAC) \wedge (SQN_{UE} < xSQN_{HN})$
If both MAC and Sync check succeed
$SQN_{UE} \leftarrow xSQN_{HN}$
$RES^* \leftarrow f_2(k, RAND, SNid)$
$K_{SEAF} \leftarrow KDF(k, RAND, SNid, xSQN_{HN})$
8.RES^*
9. $HRES \leftarrow H(RAND, RES^*)$
Check $HRES \overset{?}{=} HXRES^*$
10.$SUCI, RES^*$
11. Check $RES^* \overset{?}{=} XRES^*$
13. $GUTI$
12. $SUPI$

Fig. 5: 5G-AKA Protocol

B. The SCDHI Problem

Our schemes follow the SCDHI hardness problem defined by Camenisch et al. [22], which belongs to the family of DDH, and is a variant of the SDDHI problem [26].

Definition 1. *n-Strong Computational Diffie-Hellman Inversion Problem (SCDHI)*
Denote an oracle O^s as credential Issuance process. Thus, on input $\vec{m} = (m_1, m_2, ..., m_n)$, with $m_i \in \mathbb{Z}_p^*$, $\mathcal{O}^s$ output $\sigma = g_1^{\frac{1}{x_0 + \sum_{i=1}^{n} m_i x_i}}$.
Denote O^{d_i} as a DH operation, on input h, output h^{x_i}. The advantage is defined as:

$$\mathsf{Adv}_{n-SCDHI} = \mathsf{Pr}[(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g_1, g_2, p) \leftarrow \mathsf{Setup}(1^\lambda),$$
$$(x_0, ...x_n) \leftarrow \mathbb{Z}_p^{*n+1}, (y, m_1^*, ..., m_n^*) \leftarrow \mathcal{A}^{\mathsf{O}^s, \mathsf{O}^{d_0}, ..., \mathsf{O}^{d_n}}(g_1) :$$
$$y = g_1^{\frac{1}{x_0 + \sum_{i=1}^{n} m_i^* x_i}} \wedge (m_1^*, ..., m_n^*) \notin Q]$$

An adversary $\mathcal{A}$ who is working on a generic group of order p, and with the advantage of ϵ, will require the time of $\Omega(\sqrt[3]{\epsilon * q})$ to solve the SCDHI problem. We say SCDHI is (t, ϵ)-hard if there is no adversary that has the advantage of at least ϵ.

C. Proof of Lemma 1

We prove the LEMMA 1 in this section, and show that the proof π' in equation 4 that aims to prove knowledge of the set of BB-signatures and hidden attribute m_4 embedded in Cred is valid. During the presentation, the prover (i.e., UE) takes a random $r \in \mathbb{Z}_p^*$ and sets $\sigma' = \sigma^r$ and $\hat{\sigma_i} = \sigma_i^r$ for $i = (0, ..., 4)$. Since $\sigma_i = \sigma^{x_i}$ for $i = (0, ..., 4)$. We can obtain $\hat{\sigma_i} = \sigma_i^r = \sigma^{rx_i}$ for $i = (0, ..., 4)$.

COMPLETENESS
For $\sigma \neq 1 \in \mathbb{G}_1$, and $\sigma^{'} \neq 1 \in \mathbb{G}_1$, if a prover (UE) and a verifier (MNO) both follow the protocol execution correctly, the verifier will always accept the proof π'. The correctness of the pairing relation is as follows:

$$
\begin{aligned}
e(\bar{\sigma}, g_2) &= e(\prod_{i=1}^{4} \hat{\sigma}_i^{-m_i} g_1^r, g_2) \\
&= e(\prod_{i=1}^{4} (\sigma^{r x_i})^{-m_i} g_1^r, g_2) \\
&= e(g_1^{\frac{-\sum_{i=1}^{4} r m_i x_i}{x_0 + \sum_{i=1}^{4} m_i x_i} + r}, g_2) \\
&= e(g_1^{\frac{r x_0}{x_0 + \sum_{i=1}^{4} m_i x_i}}, g_2) \\
&= e(g_1^{\frac{r}{x_0 + \sum_{i=1}^{4} m_i x_i}}, g_2^{x_0}) \\
&= e(\sigma', X_0)
\end{aligned}
$$

SOUNDNESS

There exist a PPT algorithm that be able to extract a valid witness (r, m_4) such that it satisfy the statement in π: $\hat{\sigma}_0 \prod_{i=1}^{3} \hat{\sigma}_i^{m_i} = g_1^r \hat{\sigma}_4^{-m_4}$, which is equivalent to $\hat{\sigma}_0 \prod_{i=1}^{4} \hat{\sigma}_i^{m_i} = g_1^r$.

Since $e(\bar{\sigma}, g_2) = e(\sigma', g_2^{x_0})$, through bilinearity, we can have $\bar{\sigma} = \sigma'^{x_0}$. Given $\hat{\sigma'}_0 \prod_{i=1}^{4} \hat{\sigma'}_i^{m_i} = g_1^r$, we have $\sigma'^{r x_0} \prod_{i=1}^{4} \sigma'^{r x_i m_i}_i = g_1^r$. Thus, $\sigma' = g_1^{\frac{r}{x_0 + \sum_{i=1}^{n} m_i x_i}}$.

As $\sigma^{'} \neq 1 \in \mathbb{G}_1$, then $r \neq 0$.

A valid signature σ on the set of attributes can be extracted by computing as $\sigma = \sigma'^{1/r}$. Verifier shall be convinced that the witness (r, m_4) satisfies the relation statement in π.

ZERO-KNOWLEDGE

There exists a PPT simulator Sim, given any $r' \in \mathbb{Z}_p^*$, and sets $\sigma' = \sigma^{r'}$, $\hat{\sigma}_i = \sigma_i^{r'}$ for $i = (0, ..., 4)$. The probability distribution of the simulated proof is uniformly random.

D. *Proof of Lemma 2*

We first provide a detailed description of proving knowledge in Pres. As discussed in Section V-C, the ultimate π in Camenisch-Stadler notation is expressed as:

$$
\begin{aligned}
\pi \in ZKP\{(m_4, r) : A = g_1^r \hat{\sigma}_4^{-m_4} \wedge A = g_1^r \hat{\sigma}_4^{B} \\
\wedge -B^{-1} c_2 = h^r \wedge c_1 = g_1^r\}
\end{aligned}
$$

The statements in π only slightly differ from the statements π' that we proved in Appendix C, in fact, the statement in π is exactly the same as the sub-proof π_1 showed below in equation 6, as we let $A = g_1^r \hat{\sigma}^{-m_4}$ for obtaining succinct notations. As a result, we can prove the rest three sub-proofs similarly, as they share the same witness r and are based on the classic DH exponentiation. A detailed illustration of proving knowledge of π with a standard NIZK solution will be discussed.

E. Pres *Construction*

As an AND protocol, the proof π can be divided into four sub-proofs:

$$
\begin{aligned}
\pi_1 &\in ZKP\{(m_4, r) : A = g_1^r \hat{\sigma}_4^{-m_4}\} \\
\pi_2 &\in ZKP\{(r) : A = g_1^r \hat{\sigma}_4^{B}\} \\
\pi_3 &\in ZKP\{(r) : -B^{-1} c_2 = h^r\} \\
\pi_4 &\in ZKP\{(r) : c_1 = g_1^r\}
\end{aligned} \tag{6}
$$

The parameters within the brackets are the secret to be proven, while the rest parameters are known to the verifier, i.e., SN. The proofs are running via a non-interactive Shnorr proof with Fiat-Shamir heuristic.

Proof of π_1.
Lets $y_1 = A$. The prover, i.e., UE, executes the following steps:

1) Randomly takes $a_1, b_1 \in \mathbb{Z}_p^*$.
2) Choose challenge $C_1 = H(\bar{y_1})$
3) Compute $\bar{a_1} = a_1 + C_1 r$, and $\bar{b_1} = b_1 + C_1 m_4$
4) Sends $(y_1, C_1, \bar{a_1}, \bar{b_1})$ to the verifier,i.e., included in the Pres.

The verifier, i.e., SN then computes

$$
\begin{aligned}
\bar{y_1} &= y_1^{-C_1} g_1^{\bar{a_1}} \hat{\sigma}_4^{-\bar{b_1}} \\
&= g_1^{-C_1 r} \hat{\sigma}_4^{C_1 m_4} g_1^{a_1 + C_1 r} \hat{\sigma}_4^{-b_1 - C_1 m_4} \\
&= g_1^{a_1} \hat{\sigma}_4^{-b_1}
\end{aligned}
$$

and check if $C_1 = H(\bar{y_1})$.

Proof of π_2.
Lets $y_2 = A \hat{\sigma}_4^{-B}$. UE executes the following steps:

1) Randomly takes $a_2 \in \mathbb{Z}_p^*$.
2) Choose challenge $C_2 = H(\bar{y_2})$
3) Compute $\bar{a_2} = a_2 + C_2 r$
4) Sends $(y_2, C_2, \bar{a_2})$ to the SN.

Similar to π_1, SN then computes $\bar{y_2} = g_1^{\bar{a_2}} y_2^{-C_3} = g_1^{a_2 + C_2 r} g_1^{-C_2 r} = g_1^{a_2}$, and check if $C_2 = H(\bar{y_2})$.

Proof of π_3.
Lets $y_3 = -B^{-1} c_2$. UE executes the following steps:

1) Randomly takes $a_3 \in \mathbb{Z}_p^*$.
2) Choose challenge $C_3 = H(\bar{y_3})$
3) Compute $\bar{a_3} = a_3 + C_3 r$
4) Sends $(y_3, C_3, \bar{a_3})$ to the SN.

SN then computes $\bar{y_3} = h^{\bar{a_3}} y_3^{-C_3} = h^{a_3 + C_3 r} h^{-C_3 r} = h^{a_3}$, and check if $C_3 = H(\bar{y_3})$.

Proof of π_4.
Lets $y_4 = c_1$. UE executes the following steps:

1) Randomly takes $a_4 \in \mathbb{Z}_p^*$.
2) Choose challenge $C_4 = H(\bar{y_4})$
3) Compute $\bar{a_4} = a_4 + C_4 r$
4) Sends $(y_4, C_4, \bar{a_4})$ to the SN.

SN then computes $\bar{y_4} = g_1^{\bar{a_4}} y_4^{-C_4} = g_1^{a_4 + C_4 r} g_1^{-C_4 r} = g_1^{a_4}$, and check if $C_4 = H(\bar{y_4})$.

To make the final proof more compact, the four sub-proofs should be combined into one single proof π. The four challenges C_1, C_2, C_3, C_4 will be integrate as one **common challenge** C, by computing a hash digest over the concatenation of $\bar{y_1}$, $\bar{y_2}$, $\bar{y_3}$, $\bar{y_4}$.

By observation, the randomization parameters a_1, a_2, a_3, and a_4 can be the same value as they are all used to form the commitment over the secret r.

Let $a = a_1 = a_2 = a_3 = a_4$, thus $a' = \bar{a_1} = \bar{a_2} = \bar{a_3} = \bar{a_4}$. Which implies that, during a successful proofs, $\bar{y_2}$ should be equal to $\bar{y_4}$, i.e., a successful verification over $\bar{y_2}$ is equivalent to a verification over $\bar{y_4}$. To reduce computation cost on the UE side, we can omit $\bar{y_4}$ while computing the challenge C; instead, SN requires to perform an equality check: $\bar{y_2} \overset{?}{=} \bar{y_4}$.

Besides, the parameter nonce β should be included in the Challenge C to identify a unique Pres and prevent double-spending problems.

Therefore, the final Challenge can be expressed as

$$C = H(\bar{y_1}||\bar{y_2}||\bar{y_3}||\beta) \tag{7}$$

Let $b = b_1$, then the randomization parameters used during forming the commitment are: $a' = a + Cr$, and $b' = b + Cm_4$.

F. Pres *Verification*

Proof of π.
Lets $y_1 = A$,$y_2 = A\hat{\sigma}_4^{-B}$, $y_3 = -B^{-1}c_2$, $y_4 = c_1$.

PROOF GENERATION
UE executes the following steps:

1) Randomly takes a,$b \in \mathbb{Z}_p^*$.
2) Choose challenge $C = H(\bar{y_1}||\bar{y_2}||\bar{y_3}||\beta)$
3) Compute the response $a' = a + Cr$, and $b' = b + Cm_4$
4) Sends $(y_1, y_2, y_3, y_4, a', b')$ to the SN.
5) Output the π and the presentation Pres as:

$$\begin{aligned} \pi &= (C, a', b', y_2, y_3) \\ \mathsf{Pres} &= (\{m_i\}_{i=1}^{3}, \bar{\sigma}, \sigma', \{\hat{\sigma}\}_{i=0}^{4}, c_1, c_2, A, B, \pi) \end{aligned} \tag{8}$$

Note that, as $A = y_1, c_1 = y_4$, we omit y_1,y_4 in the Pres expression.

PROOF VERIFICATION
Once received the Pres, **SN** executes the following steps:

1) Re-compute the commitments $\bar{y_i}$:

$$\begin{aligned} \bar{y_1} &= y_1^{-C} g_1^{a'} \hat{\sigma}_4^{-b'} (= g_1^{-Cr} \hat{\sigma}_4^{Cm_4} g_1^{a+Cr} \hat{\sigma}_4^{-b-Cm_4} = g_1^{a} \hat{\sigma}_4^{-b}) \\ \bar{y_2} &= g_1^{a'} y_2^{-C} (= g_1^{a+Cr} g_1^{-Cr} = g_1^{a}) \\ \bar{y_3} &= h^{a'} y_3^{-C} (= h^{a+Cr} h^{-Cr} = h^{a}) \\ \bar{y_4} &= g_1^{a'} y_4^{-C} (= g_1^{a+Cr} g_1^{-Cr} = g_1^{a}) \end{aligned}$$

2) Compute a hash digest: $C' = H(\bar{y_1}||\bar{y_2}||\bar{y_3}||\beta)$
3) Then, **UE** performs the following equality check, and accept π as valid if all the equality holds:

$$\begin{aligned} C' &\overset{?}{=} C \\ \bar{y_2} &\overset{?}{=} \bar{y_4} \\ e(\bar{\sigma}, g_2) &\overset{?}{=} e(\sigma', X_0) \end{aligned}$$