AnonyCall: Enabling Native Private Calling in Mobile Networks

Hexuan Yu*, Chaoyu Zhang*, Yang Xiao†, Angelos D. Keromytis‡, Y. Thomas Hou*, Wenjing Lou*
*Virginia Polytechnic Institute and State University
†University of Kentucky
‡Georgia Institute of Technology

***Abstract*—Mobile Network Operators (MNOs) are known to leak or sell subscribers' sensitive information, including geolocation and communication histories. Anonymous mobile user authentication methods, such as [48] (USENIX Sec'21), [55] (NDSS'24), [13] (CCS'24), [54] (S&P'25), enable users to access mobile networks without revealing long-term identifiers like phone numbers or Subscription Permanent Identifiers (SUPI).**

However, the absence of identity transparency and location awareness poses significant challenges to implementing the above anonymous access methods in real-world mobile networks, particularly for supporting essential functions such as call routing, usage measurement, and charging. To overcome these limitations, we propose AnonyCall, a privacy-preserving call management architecture that supports anonymous mobile network access while enabling two essential functions: *anonymous callee discovery* and *usage-based charging*. The anonymous callee discovery function incorporates an out-of-band authentication mechanism to securely share temporary callee identifiers with the caller, allowing the latter to establish native calls without obtaining the callee's permanent information. The usage-based charging function introduces an anonymous and accountable balance credential that enables accurate charging and prevents double-spending while preserving mobile user anonymity. Fully compatible with existing mobile networks, AnonyCall introduces minimal overhead, adding less than 200 ms to call establishment. Evaluations with smartphones and standard calling systems demonstrate its practicality, offering a viable solution for privacy-preserving yet functional mobile communication.

I. Introduction

The evolution of mobile technologies from 2G to 5G has greatly enhanced communication services, with Mobile Network Operators (MNOs) managing infrastructure and sensitive subscriber data, including personal details, location, and call histories. However, privacy breaches and unauthorized profiling due to unethical practices, employee misbehavior, or data leaks remain significant risks, as demonstrated by major fines levied on the four major US carriers for selling subscriber data to third parties [31]. The root issue lies in MNOs' reliance on long-term identifiers like phone numbers and the Subscription Permanent Identifier (SUPI, the permanent identifier used within 5G networks), to provide services, enabling tracking and profiling by default.

Anonymous mobile network access. To address these concerns, one emerging approach is to enable users to access the mobile network anonymously, i.e., accessing network service without exposing any long-term subscription identifier (e.g., phone number, SUPI) as seen in recent solutions [51], [48], [55], [13], [42], [54]. These methods empower users to access MNO's services without disclosing their true identities (i.e., **anonymity**), and prevent the untrusted MNOs from linking multiple anonymous access sessions together (i.e., **unlinkability**). For example, [51] allows subscribers to use an ephemeral International Mobile Subscriber Identity (IMSI, the equivalent of SUPI before 5G) while using services from untrusted MNOs. [48], [55], [13], [42], [54] leverage cryptographic methods such as blind signatures and anonymous credentials to enable users to authenticate themselves to the mobile network anonymously and have unlinkable pseudonyms across different access sessions, thereby protecting phone user's identity and location privacy against untrusted MNOs. MNOs can no longer track the current location of a specific phone number nor know the real person behind a pseudonym.

A. Key Challenges and Objectives

Supporting Essential Cellular Functions amid Anonymity. Despite these advancements, there are outstanding challenges that hinder the practicality of these privacy-preserving mechanisms due to the lack of support for essential functions that are native to the incumbent mobile networks. The anonymous mobile access mechanisms as introduced in [51], [48], [55], [54] are generally designed for user access control in standalone cases, e.g., obtaining a temporary network identifier after successful anonymous authentication. They are not readily applicable (or do not specify how) to support essential cellular functions that involve continuous interaction, represented by voice calls and user charging. For example, MNOs rely on users' identifiable information, e.g., the IP addresses (allocated within the mobile core), physical location (i.e., which cell tower the user connects to), and data usage of a particular phone number, to enable voice calls and message routing between users, and billing. However, the necessity to gather such data clashes with the goals of subscriber privacy and tracking prevention. While an anonymous mobile access scheme conceals all identifiable information from the untrusted MNOs, supporting the essential mobile network functions would again require user tracking and re-identification. These constraints significantly limit the practicality of applying privacy-preserving access technologies in mobile networks. This paper aims to address the dilemmas between subscriber privacy and essential mobile network functions. Specifically, we identify two key objectives that must be

Network and Distributed System Security (NDSS) Symposium 2026
23-27 February 2026, San Diego, CA, USA
ISBN 979-8-9919276-8-0
https://dx.doi.org/10.14722/ndss.2026.241064
www.ndss-symposium.org

met by an anonymous mobile network access system:

Obj-1 : Making Anonymous User Callable through Native Mobile Networks. When using an anonymous authentication technique, the MNO can only ascertain that an anonymous but legitimate User Equipment (UE) is connected to a particular cell tower and using a specific IP address, but it cannot know this anonymous UE's phone number or SUPI (or any other long-term identifier). Meanwhile, a caller knowing the user's phone number cannot make the call through, since the callee's home network does not know which cell tower and which IP address the phone number is associated with. Therefore, the key objective is to make an anonymously authenticated user reachable to its callers. A caller does not have to be an anonymous user.

To realize the above objective, one trivial path is to use third-party Voice over IP (VoIP) services (e.g., Skype) over non-mobile networks as suggested in [48]. This, however, would require a secondary (e.g., Internet) connection and also lead to under-utilization of the mobile network's native call bandwidth and be a disincentive for the MNOs to provide anonymous service. Therefore, *it is important to enable anonymously authenticated users to use the native call functions, e.g., VoLTE and VoNR, of the same mobile network.*

Obj-2 : Real-time Balance Management and Usage Charging. In mobile networks, charging falls into two categories: *prepaid*, which requires deduction from the account balance, and *postpaid*, which involves usage recording and aggregation. Existing anonymous mobile network access schemes [48], [55], [13] do not support charging based on the UE's data usage, as explicit balance deduction or usage aggregation requires the UE to be traceable across different sessions, directly contradicting the privacy goal of unlinkability. Prepaid services face the challenge of managing real-time balances, requiring MNOs to accurately track and deduct credits across multiple unlinkable sessions tied to a single UE. Likewise, postpaid services encounter difficulties in aggregating usage across different pseudonyms without compromising anonymity or linking sessions. A *prepaid* solution proposed in [48] for charging the anonymous UE uses anonymous tokens (via blind signatures [24], [25]), where tokens represent fixed usage (e.g., one token per minute). UEs receive untraceable tokens monthly from their Home Network (HN) based on their subscription plan. However, this mechanism is unsuitable for modern mobile networks as it lacks precise, real-time usage measurement and charging. MNOs must compel UEs to pay before a session begins and allocate fixed session lengths for calls, leaving UEs unable to reclaim unused tokens if the actual call durations are shorter than the paid session. In contrast, [42] proposes a token-based *postpaid* model that relies on an independent third-party broker, requiring users to claim their consumed anonymous tokens later to settle payments. This approach introduces additional operational costs for MNOs to manage anonymous token lists and significantly deviates from traditional cellular service models. Besides, PGUS [54] considers a Thick-MVNO environment where both the MNO and MVNO are modeled as semi-honest and non-colluding. It integrates an anonymous cellular access scheme with an inter-operator billing verification mechanism using a sanitizable blind signature and a tracing technique to prevent MNO over-billing of the MVNO. While PGUS protects user anonymity and provides inter-operator accountability (i.e., MVNO-MNO billing reconciliation), it does not specify mechanisms for billing subscribers based on their individual usage. Despite these advances, existing schemes [48], [42], [54] still lack real-time, fine-grained balance management compatible with modern network operations. This gap limits the practicality of anonymous access and discourages MNO deployment. *A privacy-preserving, real-time charging framework for both prepaid and postpaid services is essential for practical adoption of anonymous cellular access.*

B. Proposed Methods and Our Contributions

Our Design. We propose ANONYCALL, a privacy-preserving call management system to enable native phone calls and usage charging while supporting anonymous mobile network access. ANONYCALL achieves the two aforementioned objectives under the condition that an established anonymous mobile network access method [48], [55], [13], [42], [54] allows a UE to anonymously authenticate itself and register with a serving MNO without revealing its permanent identifiers (e.g., phone number, IMSI/SUPI). Upon successful authentication, a UE obtains temporary identifiers from the MNO for standard mobile network access, including a dynamically assigned IP address for IP-based services within the mobile network (e.g., messaging, internet access, and call sessions), and a Uniform Resource Identifier of the Session Initiation Protocol (i.e., SIP URI) for initiating calls. It is important to note that this IP address is used solely for routing packets within the operator's infrastructure and is not publicly discoverable. The assigned temporary IP and SIP URI maintain UE unlinkability, as the MNO cannot link them to the anonymous UE's previous pseudonyms or used IP addresses.

Caller Authentication and Anonymous Callee Discovery. To achieve **Obj-1** , we introduce an out-of-band authentication and authorization method that allows an anonymous callee to authenticate the caller asynchronously and authorize the sharing of its temporary SIP URI with the caller. A SIP URI is a standardized identifier used in both cellular and non-cellular systems to uniquely identify and reach a phone user. Since the SIP URI is a standard option for dialing phone calls, the callee's home network can resolve and retrieve the callee's IP address from the SIP URI to facilitate native call routing. Integrating ANONYCALL to the existing anonymous access solution ensures that an anonymous UE remains reachable for any authenticated callers. See Sec III for details.

Privacy-preserving Usage Charging with Anonymous Accountable Credentials. To achieve **Obj-2** , we incorporate a fine-grained, privacy-preserving usage charging method facilitated by an anonymous but accountable balance credential *cred*. It leverages cryptographic techniques, including an adaptable blind signature scheme (the Structure-Preserving Signature Scheme on Equivalence Classes, or SPS-EQ), homomorphic commitments, and zero-knowledge proofs (ZKP), to support accurate charging for anonymous sessions in both prepaid and postpaid modes without revealing UE identities. For each access, a UE can transform its *cred* to an unlinkable but verifiable format, proving its validity through SPS-EQ's adaptability. The MNO can homomorphically adjust (add or subtract) the balance attribute in *cred* using its signing keys

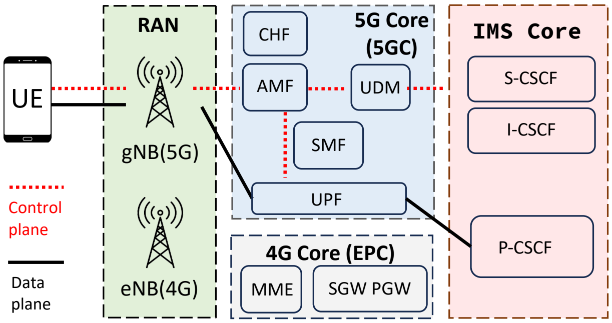


Fig. 1. A High-level Architecture of 4G and 5G Systems.
The *IMS Core*, spanning 4G LTE and 5G NR, integrates with network functions in EPC and 5GC to manage SIP-based services.
* eNB (4G LTE) and gNB (5G NR), also known as Radio Access Network (RAN), are the base stations for 4G and 5G, respectively.

without knowing the actual balance when a session ends. Several hidden tracing parameters are also embedded in the *cred*, appearing uniformly random during each session to ensure benign users' access remains unlinkable. However, if a UE attempts to reuse an old credential with a higher balance or lower cumulative usage, its identity can be automatically revealed (i.e., computed) from the tracing parameters. This approach ensures accurate and real-time charging in two charging modes without compromising the anonymity and unlinkability of benign UEs. Sec. IV details the design.

Prototype Evaluation. We deployed the two core functionalities of AnonyCall on the UE side using smartphones and evaluated the feasibility of establishing calls among anonymous UEs in a standard calling system. We measured the overhead introduced by each proposed method, and our results show that the total latency added by AnonyCall to the conventional call establishment process is below 200 ms, demonstrating that it is a viable solution for providing subscriber privacy in mobile networks.

To summarize, AnonyCall has the following contributions:

1) It enables anonymous phone users to receive calls without compromising their anonymity and unlinkability through an out-of-band authentication and temporary SIP URI sharing method.
2) It provides a fine-grained, privacy-preserving charging method that maintains the anonymity and unlinkability of benign users while de-anonymizing misbehaving UEs attempting to double-spend credentials.
3) It can be seamlessly integrated into standard calling systems native to modern mobile networks without requiring new entities or infrastructural modifications.

II. Background

A. Voice over IP and IP Multimedia Subsystem

While the conventional circuit-switch-based methods (e.g., CDMA, UMTS, GSM) for facilitating voice communications have been widely phased out in many parts of the world, VoIP, as a broader technology applicable over various network types, is increasingly becoming the standard for voice communication. VoIP is network-agnostic, meaning it does not necessarily require any specific network infrastructure beyond IP connectivity. Besides, it can function not just over the Internet but also as part of cellular networks through technologies like Voice over LTE (VoLTE), Voice over New Radio (VoNR), and Voice over WiFi (VoWiFi). VoLTE and VoNR are essentially two VoIP methods designed for delivering voice services over cellular networks. They are dependent on cellular infrastructures for functionalities like routing and connection control and require SIM-based user authentication (e.g., 5G-AKA). While VoLTE is tailored for 4G/LTE, VoNR is the 5G equivalent of VoLTE providing better bandwidth and lower latency enabled by the 5G New Radio (NR) technologies.

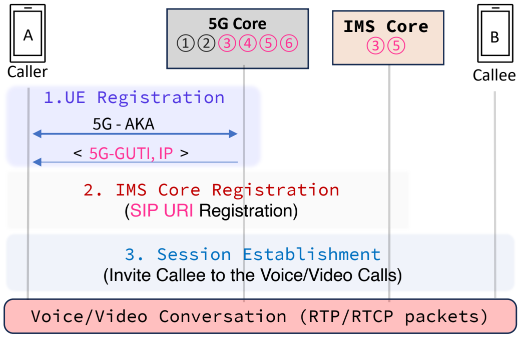


Fig. 2. A Simplified Call Establishment Flow.
* Existing anonymous authentication methods allow a UE to perform steps 1 and 2 anonymously, but they do not support step 3.
* The circled numbers are the identifiers listed in Table I.

TABLE I. Example - Subscriber *John Doe*'s Permanent and Temporary Information used within the 5G Network.
* For non-roaming cases, SN = HN.

Information	Example Format	Assigned By	Type
①**Phone Number**	`+1-870-123-4567`	HN	Permanent
②**SUPI**	`999700000000001`	HN	Permanent
③**SIP URI**	`john.doe@ims.verizon.net`	HN (IMS)	Semi-Permanent
④**5G-GUTI**	`99970000fffffff`	SN	Temporary
⑤**IP Address (IMS)**	`192.0.2.1`	HN (IMS)	Temporary
⑥**IP Address (Internet)**	`198.0.11.4`	HN/SN	Temporary

The **IP Multimedia Subsystem (IMS)** is crucial for enabling and managing IP-based multimedia services such as SMS, voice, and video calls within cellular networks. Fig. 1 illustrates the IMS core integration in both 4G and 5G architectures, with VoLTE and VoNR relying heavily on it for multimedia services over IP networks. Non-cellular VoIP providers often use simpler architectures containing some IMS elements. The **Session Initiation Protocol (SIP) signaling protocol** is integral for both cellular and non-cellular VoIP services, handling session management tasks such as establishing, modifying, and terminating multimedia sessions, no matter if they deploy the whole IMS core. On mobile devices, the IMS client for VoLTE and VoNR services is typically embedded in firmware by manufacturers or chipset vendors. The IMS stack includes the necessary components like the SIP for signaling, and the Real-time Transport Protocol (RTP) for media transmission, integrating with the operating system to deliver multimedia services over cellular networks.

B. Call Establishment Flow in 5G Networks

This section outlines how a UE attaches to the 5G and IMS cores and initiates calls (Fig. 2), with corresponding subscriber identifiers and additional details provided in TABLE I.

1. AMF Registration and IP Assignment. To access any

services on a mobile network, a UE must first register with the Access and Mobility Management Function (AMF) in the 5G Core (5GC) or the Mobility Management Entity (MME) in the Evolved Packet Core (EPC) for 4G. This registration occurs in the control plane, as depicted in Fig. 1, by initiating the authentication and key agreement process (i.e., the 5G-AKA protocol). This process relies on *subscription credentials*, including SUPI and the permanent keys pre-loaded into the subscriber's SIM card. The SIM acts as the root of trust in cellular networks, and the combination of the SUPI and permanent keys verifies the subscriber's identity to the MNO.

After successful authentication, the AMF assigns the UE a fresh 5G-Globally Unique Temporary Identifier (5G-GUTI), indicating successful registration and granting temporary access. The UE then establishes the necessary PDU sessions for service connectivity. A typical UE maintains at least two active IP addresses per registration, one for internet access and one for IMS services, with additional addresses possible for services such as VPNs. The Session Management Function (SMF) coordinates with the User Plane Function (UPF) to allocate these addresses:

- an *Internet IP address* from a general pool or obtained dynamically via the operator's DHCP server;
- an *IMS IP address* from a dedicated IMS IP pool or via DHCP in the IMS domain.

The SMF also instructs the UPFs to establish user-plane paths to carry traffic associated with the UE's assigned IP addresses.

During Roaming. UE registers with the AMF of the serving network (SN), which authenticates the UE by contacting the UE's HN. The SN determines the 5G-GUTI for the roaming UE. In most deployments, the *Internet IP address* is allocated by the HN's UPF (*home-routed roaming*); in less common *local breakout* scenarios, it may be assigned by the SN. The *IMS IP address*, however, is **always** allocated by the HN, even when the UE is roaming, and the SIP traffic is routed to the P-CSCF in the HN. This ensures policy and charging control and support service continuity (e.g., seamless call handover).

AMF Re-registration and IP Re-assignment Interval. The Re-registration interval usually ranges from a few hours to a couple of days and is determined by different MNOs, besides, it will also be triggered by events such as device power-off, SIM removal, or airplane mode. This periodic update ensures that the UE remains connected and reachable via the 5GC. As noted in [38], [55], the 5G-GUTI should be refreshed frequently to enhance unlinkability, limiting the duration for which a UE retains the same identifier. In addition, a new IP address is generally assigned when the underlying PDU sessions are released or expire. The actual reassignment interval depends on operator-specific configurations and session management policies.

2. IMS Core Registration. To enable VoLTE or VoNR services on the data plane, a UE must register with the IMS core by sending a `SIP REGISTER` request to the Proxy Call Session Control Function (P-CSCF) in its home IMS core. This request includes the UE's IMS Public User Identity (IMPU), typically a phone number and/or a SIP URI — an email-like address used for identifying callers and callees during call setup. A phone number maps to one or more SIP URIs in the IMS architecture. For example, a Verizon subscriber, John Doe, with the phone number `+1-870-123-4567`, can have SIP URIs like `sip:18701234567@ims.verizon.net` and `sip:john.doe@ims.verizon.net`. These SIP URIs, along with other UE information (e.g., IP, 5G-GUTI, Cell ID), are stored in the Unified Data Management (UDM) for 5G or the Home Subscriber Server (HSS) for 4G. The Serving Call Session Control Function (S-CSCF) authenticates the UE by interacting with the UDM through an AKA process (similar to 5G-AKA). After authentication, the UDM provides the S-CSCF with the user's subscription information, such as the subscribed plan and allowed usage. The S-CSCF finalizes registration by sending a `SIP 200 OK` response to the UE.

During Roaming. Unlike AMF registration, the UE registers only with its home IMS core, and as noted earlier, the IMS IP is always allocated by the HN, even when roaming. To access IMS-based services such as voice calls, the UE connects to the HN's IMS core through the SN's infrastructure.

IMS Re-registration Interval. This interval is determined by the expiration timer in the `SIP REGISTER` response. Common values are 10 or 60 minutes, but it can be adjusted based on the policies and requirements of individual MNOs.

3. Session Establishment. When the UE initiates a call, the IMS converts the callee's phone number into a SIP URI via ENUM or DNS lookup. A `SIP INVITE` request, containing the callee's and caller's SIP URIs along with the caller's current IP address, is sent to the callee's home IMS core. Upon a successful handshake—meaning the callee accepts the call and step 3 in Fig. 2 is completed—voice or video data packets are exchanged directly between the caller and callee using the RTP and Real-time Transport Control Protocol (RTCP) over the established IP connection.

During Roaming. To ensure authentication, billing, and policy enforcement, all call session establishments *must* traverse a user's home IMS core, even while roaming. For instance, consider caller **A** from carrier **1** and callee **B** from carrier **2**. If caller **A** is not roaming but callee **B** is roaming on carrier **3**, **A**'s `SIP INVITE` is routed via **1**'s IMS core, then to **B**'s home IMS core (carrier **2**), and finally to the visited IMS of carrier **3**. If **B** accepts the call, the media path is established directly between carriers **1** and **3**.

Charging Methods. In 5G, the Charging Function (CHF) manages subscribers' charging records. The SMF tracks session data usage and reports it to the CHF. For *prepaid* accounts, the CHF authorizes service usage based on the subscriber's balance, authorizes usage based on predefined thresholds, and deducts credit in real-time, which is also known as *real time online charging (OCS)* in 5G's Converged Charging System (CCS). For *postpaid* accounts, the CHF collects usage data records per session. These records are then aggregated later for billing (i.e., *batch-based offline charging*).

During roaming. As all IMS-based services (e.g., voice calls) are *home-routed*, the HN maintains full visibility and control over each session, even during roaming. This allows the HN to accurately charge the user based on real-time session data. In contrast, for services utilizing *local breakout*, the HN does not share the UE's account balance with the SN. In prepaid scenarios, the SN monitors session activity and periodically requests authorization from the HN. In postpaid scenarios, the

SN generates charging records and forwards them in batches to the HN. In some cases, the SN bills the HN based on wholesale agreements, without enforcing user-specific usage limits.

C. Telecom PKI and Global Identity Frameworks

Our out-of-band authentication method establishes caller-callee trust without new authorities. While AnonyCall does not assume any specific PKI, it requires a minimal trust anchor that can be instantiated using existing frameworks already deployed in telecom and digital-identity ecosystems. For example, on the telecom side, systems such as *STIR/SHAKEN* [30], mandated across U.S. carriers and adopted in regions such as France, the U.K., and Canada, show how operator PKI is formed: each carrier maintains its own signing keys and obtains certificates from accredited CAs, enabling cross-domain verification without private-key sharing. Trust and *revocation* follow standard CA hierarchies (CRLs/OCSP), and the same model applies in *roaming* where operators validate identity information using their own certificates. In parallel, the W3C-standardized *Decentralized Identifiers (DIDs)* [2] and *Verifiable Credentials (VCs)* [12] provide portable identity credentials that can be verified globally. These standards underpin major national and industrial frameworks, including the EU's eIDAS regulation [3], and the EU Digital Identity Wallet (EUDI Wallet) [26], Australia's TDIF [1], Canada's PCTF [11], Microsoft Entra Verified ID [10], and the Linux Foundation's CREDEBL project [52]. In these systems, users hold digital wallets containing VCs issued by trusted authorities, and global interoperability arises because each VC is verified using the issuer's public keys and revocation data published in public DID registries, rather than any single national certificate hierarchy. The standards also support zero-knowledge proofs for selective disclosure, such as proving age eligibility without revealing a full birth date or address.

Together, these deployments show that cross-jurisdiction trust can be achieved without shared secrets or new authorities. AnonyCall can therefore rely on these existing infrastructures for secure, standard-compatible trust establishment while remaining agnostic to any particular identity system.

III. Caller Authentication and Anonymous Callee Discovery

Overview. This section introduces our first component, which enables native calling to anonymous callees. AnonyCall assumes that a UE can anonymously register with the 5G and IMS cores via existing solutions [48], [55], [13], [42], [54] without revealing its phone number or SUPI. To make an anonymous callee reachable by callers, we incorporate an out-of-band authentication method (Sec III-A) to help the anonymous callee establish secure channels and share its temporary SIP URI with trusted callers, serving as the preamble for call initiation. The caller can then initiate a private call with the callee using the obtained temporary SIP URI via the native SIP process within mobile networks (Sec III-B).

Threat Model. AnonyCall assumes that MNOs are honest but curious. They operate cellular network functions as expected, adhering to protocols such as allocating dynamic IP addresses via DHCP for registered UEs, routing calls correctly between callers and callees, and accurately executing charging protocols and billing users based on actual usage. However, MNOs may attempt to compromise privacy by uncovering the permanent identities (e.g., phone numbers, SUPIs) of anonymous UEs or identifying participants in specific conversations.

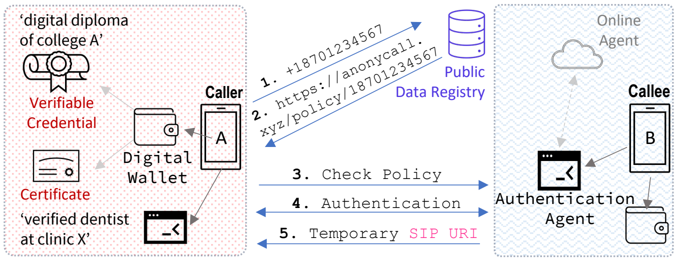


Fig. 3. Out-of-Band Caller Authentication
* Steps 1–4 are required only for the first call between a given caller–callee pair; subsequent calls can skip re-authentication.
* Each UE has its own *digital wallet* and *authentication agent*.

A. Out-of-Band Caller Authentication

High-level Workflow. As shown in Fig. 3, the caller first dials the callee's phone number (Step 1) and is directed to the callee's authentication agent URL (Step 2), for example via a prerecorded greeting, or a DID-based record. The caller then authenticates with the agent according to the callee's policy (Steps 3-4). Once authorized, the agent returns the callee's temporary SIP URI (Step 5). The caller then re-dials using this SIP URI, and the IMS completes the call through the standard SIP routing process without exposing any long-term identifiers (to describe in Sec. III-B).

a. MNO side - Temporary SIP URI Assignment during Anonymous UE Registration. The out-of-band authentication function requires minimal changes on the MNO side. Specifically, as introduced in Sec II-B, the home IMS core typically registers the UE's SIP URI every 10 or 60 minutes. Our method slightly modifies this process by having the home IMS core assign and register a fresh SIP URI for an anonymous UE at this step (i.e., step 2 of Fig. 4), based on the UE's IP address (for IMS) assigned by the 5GC (SMF) during UE Registration (step 1). This is feasible because a UE can have multiple SIP URIs simultaneously in current cellular networks. A temporary SIP URI is unique within the operators' domain as long as it correctly binds to the UE's current IMS IP address, e.g., a UE with `IP:192.0.2.1` allocated in step 1 of Fig. 4 can immediately obtain a SIP URI `sip:192.0.2.1@ims.anonycall.xyz` from its IMS core at step 2. This SIP URI becomes invalid once the IP address is released, e.g., indicated by a `DHCP RELEASE` message sent by the UE. This method ensures that anonymous callee's SIP URI always maps to its real-time IMS IP address and there are no SIP URI collisions within the mobile network. As SIP URI natively supports appending optional and customizable headers and parameters for conveying extra details, we can define a privacy parameter to indicate the call mode, e.g.:

Parameter: privacy = <level >
Description: Define a privacy level for the call
Example: `sip:192.0.2.1@ims.anonycall.xyz;`
`privacy=full`

This parameter instructs the IMS core to resolve a SIP URI to its IMS IP address by using the content before the @ symbol.

While dialing through SIP URI is a standard method supported by many applications (discussed by **Q1** in Sec V-B), this approach guarantees discoverability whenever someone dials an anonymous UE's temporary SIP URI.

b. UE side - A Caller Authentication Agent for Sharing the Temporary SIP URI. Figure 3 illustrates a high-level out-of-band authentication flow between a callee and an unknown caller (i.e., the callee has no prior knowledge of the caller). We consider both the *digital wallet* on the UE, which stores VCs or digital certificates, and the *authentication agent* on the UE to be trustworthy, as they are usually deployed as mobile apps and operate within the same security domain as the user. Scenarios involving collusion between the agent and an MNO are excluded from consideration. Specifically, when a UE acts as a caller, it retains full control over its digital wallet and voluntarily consents to disclose VCs or certificates for authentication when requested by the callee's agent. Conversely, when the UE is the callee, its authentication agent will only share sensitive information (i.e., a temporary SIP URI) with an authenticated caller. To facilitate the authentication of unknown callers and the controlled sharing of the temporary SIP URI, a UE can predefine an *authentication policy* via its authentication agent. This policy defines the conditions under which the agent will disclose the UE's current SIP URI to a caller. The authentication agent app on the UE periodically retrieves the current SIP URI by accessing the SIP information stored on the UE's IMS client[1].

Authentication Policy and Methods. The policy can be as simple as automatically allowing any contact stored in the callee's contact book or any registered business number listed on Google to pass authentication. More flexible policies can be implemented by leveraging existing trust frameworks mentioned in Sec II-C. For instance, the callee (say, John) can configure the authentication agent to accept calls from authorities, healthcare providers, and classmates. The agent will then authorize sharing the SIP URI with a caller who presents a certificate or VC with valid digital signatures, demonstrating that they satisfy the specified criteria, such as being a verified doctor or graduating from the same college as John. Crucially, the authentication agent *does not* require John to operate manually or to be always online. It runs as a highly available web service on behalf of John—automatically authenticating and sharing John's current SIP URI with the callers when they satisfy the policies. An optional *Online Agent* (e.g., a cloud-based service) can be employed to periodically synchronize the SIP URI and handle authentication when John's device has limited network access. The secure storage of the SIP URI in a cloud-based agent depends on standard cloud security measures, which are beyond the scope of this paper. Besides, the temporary SIP URI can optionally be hidden from the authentication and online agents by using a common client-side encryption model (e.g., Apple's iCloud Keychain [8]), especially when the service is deployed by a semi-trusted party such as a dedicated Mobile Virtual Network Operator (MVNO). We further discuss this scenario in Q3, Sec. V-B.

After successful authentication, the caller can use the obtained SIP URI to call the anonymous callee, John. John's home IMS core can resolve the SIP URI to an IP once it notices the `Privacy` parameter in the SIP header, and route the call as usual. However, it cannot find out the callee's identity as John previously registered anonymously. Additionally, callers *do not* need to authenticate with John's agent before every call. Authentication can be performed once, after which John's agent grants the caller access rights to the SIP URI for an extended period (e.g., one year). This allows the caller to retrieve John's most recent SIP URI without repeating the authentication process for each call.

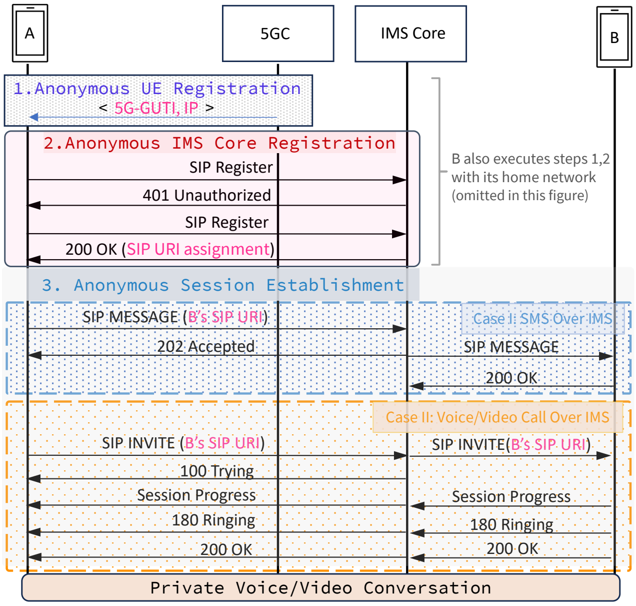

A
5GC
IMS Core
B
1.Anonymous UE Registration
< 5G-GUTI, IP >
2.Anonymous IMS Core Registration
SIP Register
401 Unauthorized
SIP Register
200 OK (SIP URI assignment)
B also executes steps 1,2 with its home network (omitted in this figure)
3. Anonymous Session Establishment
SIP MESSAGE (B's SIP URI)
Case I: SMS Over IMS
202 Accepted
SIP MESSAGE
200 OK
Case II: Voice/Video Call Over IMS
SIP INVITE (B's SIP URI)
SIP INVITE(B's SIP URI)
100 Trying
Session Progress
Session Progress
180 Ringing
180 Ringing
200 OK
200 OK
Private Voice/Video Conversation

Fig. 4. Anonymous and Private Call Establishment Flow

Q: *How can a caller know the callee is in the anonymous mode and reach the callee's Authentication Agent?* An anonymous UE can simply play a prerecorded greeting or voicemail indicating that it is operating in anonymous mode and directing the caller to its authentication agent. A more flexible option is to leverage the DID infrastructure: per W3C standards, each entity can publish a DID document in a public registry (e.g., a ledger). This JSON-formatted document includes a unique identifier, verification information (e.g., encryption algorithms and public keys), and a `ServiceEndpoint` field linking to specific services. In AnonyCall, John's DID document can include the entry that links to its authentication agent, e.g., "`ServiceEndpoint`": "`https://anonycall.xyz/policy/18701234567`". This URL embeds a phone number, allowing the caller to reach the authentication agent as described earlier (steps 1 and 2, Fig. 3).

B. Private Call Establishment Flow

Fig. 4 shows the full process of establishing an anonymous and private call within the mobile network. Although the example assumes both **A** and **B** are anonymous, AnonyCall *does not* need a caller to be anonymous, i.e., **A** can register with the mobile network through standard method. Our primary focus is on how to direct calls to an anonymous callee **B**.

Anonymous Registration with 5GC and IMS core. **A** and **B** anonymously register with 5GC and IMS core, obtaining

[1] For example, the `TelephonyManager` and `SubscriptionManager` classes in Android allow developers with proper permissions to access the UE's IMS profile, including the carrier-assigned SIP URI.

a dynamic IMS IP address (step 1) and a temporary SIP URI (step 2). The temporary SIP URI is assigned within the standard SIP message `200 OK`, sent from the UE's home IMS core, indicating successful SIP registration with the IMS service. ANONYCALL does not introduce additional steps to the standard SIP process; steps 2 and 3 illustrate the usual SIP registration and session establishment protocol. The only change is that the SIP URI embedded in the header of the SIP messages `200 OK`, `SIP MESSAGE`, and `SIP INVITE` is now a temporary SIP URI instead of a semi-permanent one.

Anonymous Session Establishment (SIP Signaling). In a standard SIP session establishment process, the header of a `SIP MESSAGE` (for SMS) or `SIP INVITE` (for voice/video calls) includes both the caller's and callee's SIP URIs. In ANONYCALL, the temporary SIP URI enables the IMS core to resolve the anonymous callee's IP address and route the SIP message accordingly without modifying the SIP signaling process. When **A** calls **B**, instead of dialing **B**'s phone number, **A** dials the temporary SIP URI of **B**, obtained via the out-of-band authentication method (how to dial SIP URIs is explained in Q1, Sec V-B). After a successful handshake between **A** and **B**, communication is routed via the RTP protocol between their IP addresses. Besides, ANONYCALL remains effective during *roaming*, as discussed in Q3, Sec V-B.

IV. PRIVACY-PRESERVING CHARGING METHOD

This section introduces ANONYCALL's second core component — privacy-preserving charging for anonymous UEs. We begin with a high-level overview, introduce essential cryptographic preliminaries, and detail the charging process.

Threat Model. The *honest-but-curious MNO* assumption in Sec III applies here, consistent with [48], [55], [13], [42], [54]. Specifically, a curious MNO may try to identify the UE of a particular $cred$ or link multiple credentials to the same UE. Additionally, we assume a dishonest UE may attempt *double spending attacks* by reusing previous credentials with a higher balance or lower cumulative usage.

A. Overview

The UE obtains a *balance credential* ($cred$) at the start of each month, typically following payment. During issuance, the UE *is not* anonymous, as the HN must verify its payment status and subscription plan. Users are thus expected to complete payment to receive or renew the $cred$, which then enables unlinkable charging and usage measurement.

Note that, unlike VCs, which are bootstrapped by existing W3C infrastructure and held in digital wallets, $cred$ is issued by the UE's HN using standard cellular practices such as Over-the-Air (OTA) SIM provisioning.

To enable usage-based charging while preserving UE privacy, our $cred$ design leverages a malleable and adaptable signature scheme (SPS-EQ) to construct a balance attribute. Specifically, $cred$ includes an attribute indicating allowed usage (prepaid) or aggregated usage (postpaid), secret parameters (e.g., tracing index) for double-spending detection, and a signature over the (committed) attributes. Critically, SPS-EQ's properties allow the UE to adapt an original signature-message pair into a fresh, unlinkable but valid pair ($cred'$), ensuring the HN cannot trace $cred'$ to its original signed form.

During network access, an anonymous UE avoids revealing its actual account balance to prevent the HN from tracing its activity. Instead, it uses a commitment scheme to conceal the balance attribute and applies ZKP to demonstrate eligibility for services based on its plan type. This enables the HN to grant network access without creating linkability.

To be more specific, in **prepaid** mode, the UE can prove its hidden balance $\geq$ a threshold th predefined by the operator (e.g., $th = 5$ minutes) via ZKP (e.g., a range proof), allowing SMF to authorize a 5-minute slot. If the allocated usage time is running low during the session (e.g., if the UE has already consumed 4 minutes), the SMF can instruct the UE to prove again that its balance exceeds a larger amount (e.g., ≥ 10 minutes). The SMF tracks call time and reports actual usage to the CHF, which deducts it from the hidden balance using the Pedersen Commitment scheme's homomorphic properties without accessing the UE's actual balance. The CHF then generates a new signature for the updated balance, enabling the UE to receive an updated credential. Note that such a predefined threshold th is common in carrier deployments and may vary with service requirements or MNO policy. To prevent link anonymous sessions (e.g., inference attacks via binary search), we assume th is fixed by prior agreement with the MNOs and applied uniformly to all users (e.g., 1-, 5-, or 10-minute credit), ensuring that proof behavior does not reveal per-UE information.

In **postpaid** mode, the process is simpler, as the UE does not need to prove its balance before the session. Instead, the CHF adds the actual usage to the UE's hidden cumulative balance at the end of the session and updates its credential. This approach allows the CHF to manage prepaid deductions or postpaid usage for an anonymous UE while preserving anonymity and unlinkability.

Roaming. As discussed in Sec. II-B, IMS activity remains fully visible to the HN even during roaming, since all IMS traffic is *home-routed* through the HN's IMS core. In both prepay and postpay scenarios, the SN is only responsible for establishing the IMS PDU session with QoS parameters provided by the HN, and neither the SN nor the HN learns the UE's identity.

Double Spending Detection. We incorporate a double-spending detection mechanism, commonly used in offline e-cash systems, to prevent UEs from reusing credentials with higher balances or lower cumulative usage. ANONYCALL purposefully provides *conditional unlinkability* by embedding the UE's identity (specifically, the `MSIN` field from the SUPI, which uniquely identifies a subscriber within a carrier's database) into $cred$, along with two secret parameters that uniquely bind to each credential version. These parameters remain hidden and appear random as long as the UE uses each $cred$ only once with the same balance. If double-spending occurs, the misbehaving UE's identity can be de-anonymized, discouraging the reuse of old credentials.

B. Cryptography Building Blocks

1) Zero-Knowledge Proof (ZKP): ZKP allows a prover to demonstrate knowledge of a secret without revealing it.

Schnorr's protocol [49] is a fundamental ZKP method where the prover shows knowledge of a secret $x \in \mathbb{Z}_p$ such that $h = g^x \mod p$, relying on the hardness of the discrete logarithm problem (DLP). Schnorr's protocol is a specific instance of a three-round Σ-protocol [28], which can be converted into a non-interactive zero-knowledge proof (NIZK) using the Fiat-Shamir heuristic [32] under the Random Oracle Model (ROM) [50]. We use the Camenisch-Stadler notation [22] to represent the ZKP of discrete logarithms and concurrent statements throughout the paper. An illustrative notation is:

$$\pi \in ZKP\{(\alpha, \beta) : y = g^\alpha h^\beta \wedge \alpha \in [0, 2^n - 1]\} \quad (1)$$

The parameters preceding the colon are the secrets only known to the prover, while the rest are public and known to both the prover and verifier. This AND-composition example represents a ZKP of knowledge for the secrets α and β such that $y = g^\alpha h^\beta$ and α lies within a range $[0, 2^n - 1]$, where g and h are elements of a group $\mathbb{G}$ with prime order p, n is the number of bits used to represent the bounded range. This setup allows the prover to prove the correctness of the statement $y = g^\alpha h^\beta$ without disclosing α and β. A range proof (e.g., [21], [20]) that relies on commitment schemes and NIZK can be used to prove α is within a certain range without revealing any additional information about α. By convention, π outputs a boolean value indicating the validity of the proof.

2) Generalized Pedersen Commitment: Pedersen commitment [43] is a scheme that allows one to commit to a chosen value while keeping it hidden and revealing it later. Its homomorphic properties enable operations like addition directly on committed values, making it useful for privacy-preserving protocols. ANONYCALL uses the Generalized Pedersen Commitment, which allows committing to multiple messages or attributes simultaneously. Additionally, it can also integrate seamlessly with ZKP, enabling a prover to prove knowledge of committed values without revealing them.

Setup. Choose a large prime p and let $\mathbb{G}$ be a cyclic group of prime order p. Select n generators $g_1, g_2, \dots, g_n$ and h from the group $\mathbb{G}$ and publish them.

Commit. To commit to a vector of values $\vec{m} = (m_1, m_2, \dots, m_n) \in \mathbb{Z}_p$, generate a blinding factor $r \xleftarrow{R} \mathbb{Z}_p$, compute the commitment as $C = g_1^{m_1} \cdot g_2^{m_2} \cdots g_n^{m_n} \cdot h^r$. The commitment C has two properties: (1) ***Hiding***. It conceals the vector $\vec{m}$ due to the randomness introduced by r; (2) ***Binding***. It is computationally infeasible to find another vector $\vec{m}'$ and r' that produce the same commitment C.

Homomorphism. Pedersen commitment is additively homomorphic, enabling addition over committed messages m_i without knowing the $(m_1, m_2, \dots, m_n, r)$. For example, given a new message m_1', an updated commitment can be computed as $C' = C \cdot g_1^{m_1'} = g_1^{m_1+m_1'} g_2^{m_2} \dots g_n^{m_n} h^r$. The C' is in fact a commitment over $(m_1 + m_1', m_2, \dots, m_n)$. This property allows efficient and secure updates to committed values, e.g., the balance attribute of a UE's *cred*.

3) Structure-Preserving Signature Scheme on Equivalence Classes (SPS-EQ): SPS-EQ is a malleable signature scheme that is existentially unforgeable under a chosen message attack (*EUF-CMA*) in bilinear group environments (Type-III pairings). It allows the same message to be represented in multiple ways while still being considered equivalent (explained in Appendix A-A2), making it ideal for constructing privacy-preserving protocols [34], [19], [35], [27].

Adaptability. Given a signature σ on a message M, SPS-EQ enables a user to transform σ into a fresh signature σ' for another representation $M' = M^s$, with a scalar $s \xleftarrow{R} \mathbb{Z}_p^*$ without knowing the signing keys. The new σ' and M' are indistinguishable from the originals, ensuring the signer cannot link σ' to σ or M' to M. Besides, any party can verify that σ' is a valid signature of the new representative M' of the message M. Details of the SPS-EQ scheme and its adaptability are provided in Appendix A-A. For simplicity, we use abstract functions in the following discussions. Specifically, $\sigma = \mathsf{Sign}(sk_{HN}, M)$ denotes the signing process, where the signature σ is generated for M using the HN's secret key sk_{HN}. Similarly, $\mathsf{Verify}(M', \sigma', pk_{HN})$ represents the verification process, where any party can validate the adapted signature σ' for M' using the HN's public key pk_{HN}.

C. Protocol Details

We now explain the details of the charging method. The full protocol workflow is shown in Fig. 7 in the Appendix.

1) Setup and Key Generation. : This is a one-time process in which the HN generates the parameters required to issue *cred* for all its subscribers. The public parameters can be distributed to all UEs, for instance, via OTA SIM provisioning.

$\underline{\mathsf{Setup}(1^\lambda)}$. Given a security parameter λ, output a bilinear group BG. (Details on the bilinear group are not essential for understanding our scheme and are deferred to Appendix A-A1).

$\underline{\mathsf{KeyGen}(\mathsf{BG}, \ell)}$. On input BG and a vector length $\ell = 2$, this algorithm outputs a secret-public key pair (sk_{HN}, pk_{HN}) for the HN, which is used for issuing the *cred* for the UE, i.e., generating the SPS-EQ signature on a certain message. Then it outputs the components for the generalized Pedersen commitments: select $x_i \xleftarrow{R} \mathbb{Z}_p^*$, and set $h_i = g_1^{x_i}$ for $i = 0, \dots, 3$. Only HN knows the *secret* trapdoors $\{x_0, x_1, x_2, x_3\}$, while g_1 and $\{h_0, h_1, h_2, h_3, h_4\}$ are *public* parameters known to UEs.

2) Balance Credential Issuance and Obtain: This process includes steps ①②③. (The upper portion of Fig. 7 depicts the whole process.) To obtain a *cred*, the UE first prepares four parameters that will be embedded within the *cred*, where k, id, d are used for double-spending detection:

- M, an updatable *balance attribute* initialized by the UE as $M_0 = 0$, with its actual value determined later by the HN. In prepaid plans, M represents the *allowed balance* (e.g., M minutes of voice calls per month), and the HN blindly adds the allocated amount based on the UE's plan monthly. In postpaid plans, M tracks *cumulative usage*.
- k, a *hidden identity attribute*, also initialized as $k_0 = 0$, and later set by HN (i.e., MSIN). It can be used to de-anonymize the UE during double-spending.
- $id \xleftarrow{R} \mathbb{Z}_p$, a random number serving as the *cred* index.
- $d \xleftarrow{R} \mathbb{Z}_p$, a secret random number.

① UE commits to the initialized parameters.
$\underline{\mathsf{Commit}(M_0, k_0, id, d, r)}$. UE computes a Pedersen commitment of M_0, k_0, id, and d, where r is the blinding factor:

$$C = h_0^{M_0} \cdot h_1^{k_0} \cdot h_2^{id} \cdot h_3^{d} \cdot h_4^{r}$$

Next, UE forms (C, g_1), where g_1 is a public parameter. At this stage, the UE is *not* anonymous, as the HN must verify the subscribed plan to determine M and k for the credential. To ensure anonymity and unlinkability in future credential use, the UE randomizes the message to be signed by transforming (C, g_1) into $(C^{S_0}, g_1^{S_0})$ with a randomizer $S_0 \in \mathbb{Z}_p$.

Additionally, since M_0 and k_0 must be initialized to 0 by the UE to allow the HN to set the correct values for M and k, the UE proves the correctness of C^{S_0} by computing a ZKP π_0, which implicitly proves $M_0 = 0$ and $k_0 = 0$:

$$\pi_0 \in ZKP\{(M_0, k_0, id, d, r, S_0) : C^{S_0} = (h_2^{id} \cdot h_3^{d} \cdot h_4^{r})^{S_0}\}$$

UE then sends $(C^{S_0}, g_1^{S_0})$ and the proof π_0 to the HN. (Detailed proofs for all ZKPs, π_0, π_1, and π_2 are in Appendix A-B.)

② $cred$ **Issuance by HN - Load the initial account balance** M**, add the secret identity** k**, and generate a signature** σ**.** $\mathsf{Issue}((C^{S_0}, g_1^{S_0}), \pi_0, x_0, x_1, sk_{HN})$. For a *prepaid* user, the HN needs to set the allowed usage M based on the user's plan and set the exponent k as the user's `MSIN`. With knowledge of the secrets x_0, x_1, HN can homomorphically add M and embed k without knowing the actual values of the exponents in C^{S_0}. This is accomplished by computing:

$$\begin{aligned} C_0^{S_0} &= C^{S_0} \cdot g_1^{S_0 \cdot x_0 \cdot M} \cdot g_1^{S_0 \cdot x_1 \cdot k} \\ &= (h_0^0 \cdot h_1^0 \cdot h_2^{id} \cdot h_3^{d} \cdot h_4^{r})^{S_0} \cdot h_0^{S_0 \cdot M} \cdot h_1^{S_0 \cdot k} \qquad (2) \\ &= (h_0^M \cdot h_1^k \cdot h_2^{id} \cdot h_3^{d} \cdot h_4^{r})^{S_0} \end{aligned}$$

For *postpaid* user, the M is initialized as 0, HN only needs to embed k by using the same method and outputs $C_0^{S_0} = (h_0^M \cdot h_1^k \cdot h_2^{id} \cdot h_3^{d} \cdot h_4^{r})^{S_0}$, in which $M = 0$.

Given the randomized commitment vector $(C_0^{S_0}, g_1^{S_0})$, HN generates a SPS-EQ signature $\sigma \leftarrow \mathsf{Sign}(sk_{HN}, (C_0^{S_0}, g_1^{S_0}))$, then sends σ, $(C_0^{S_0}, g_1^{S_0})$ to UE.

③ **UE obtains the initial credential** $cred_0$**, and prepares new secret parameters for its successor credential** $cred_1$**.** $\mathsf{Obtain}((C_0^{S_0}, g_1^{S_0}), \sigma, \frac{1}{S_0}, pk_{HN}, x_1)$. The UE first unblinds the received $(C_0^{S_0}, g_1^{S_0})$ to (C_0, g_1) using its secret S_0. It then adapts σ to σ_0, obtaining an unlinkable message-signature pair by using SPS-EQ's adaptability through the algorithm $\mathsf{Adapt}((C_0^{S_0}, g_1^{S_0}), \sigma, \frac{1}{S_0}, pk_{HN})$, and outputs $((C_0, g_1), \sigma_0)$ (the algorithm $\mathsf{Adapt}(\cdot)$ can be found in Appendix A-A2). The *initial balance credential* for a UE is

$$cred_0 : ((C_0, g_1), \sigma_0)$$

It consists of the commitment (C_0, g_1) and the signature σ_0. A UE can use $cred_0$ without being linked to $C_0^{S_0}$, the value originally signed by the HN.

To support double-spending detection, each updated $cred$ must have a unique id and d. Therefore, the UE generates new secret parameters for its successor credential $cred_1$ to ensure uniqueness. Given the same M and k, the UE generates id', $d' \xleftarrow{R} \mathbb{Z}_p$, and a blinding factor r', forming the commitment:

$$C_1 = h_0^M \cdot h_1^k \cdot h_2^{id'} \cdot h_3^{d'} \cdot h_4^{r'}$$

Then, it randomizes C_1 to $C_1^{S_1} = (h_0^M \cdot h_1^k \cdot h_2^{id'} \cdot h_3^{d'} \cdot h_4^{r'})^{S_1}$ as in eq.(2), and stores $(C_1^{S_1}, g_1^{S_1})$. In C_1, the balance attribute M and the hidden identity attribute k must match those in C_0 of the current credential $cred_0$. Thus UE must also generate a proof to *prove their equality*, which can be pre-computed offline immediately after a session ends, as the new $cred$ with the updated M is issued at that time. This approach eliminates the overhead introduced to the call session setup phase.

3) Use $cred$ *during Session Establishment:*

UE's Tasks. During network access, a *prepaid* **UE** needs to:

1)Prove the $cred_0 = ((C_0, g_1), \sigma_0)$ has not been double-spent.
2)Prove $((C_0, g_1), \sigma_0)$ is a valid message-signature pair.
3)Prove it has sufficient balance to start a session without revealing the actual balance M to the HN by showing proof that $M \geq th$, where th is a constant pre-defined by the HN.
4)Send the pre-computed $(C_1^{S_1}, g_1^{S_1})$ to the HN to form the successor credential $cred_1$ and provide proof that M and k in $C_1^{S_1}$ match the corresponding exponents in $cred_0$.

HN's Tasks. Accordingly, the **HN** needs to:

1)Check if $cred_0$ has been double-spent,
2)Verify the authenticity of the signature σ_0 within $cred_0$.
3)Verify a range proof of M.
4)Verify that the M and k embedded in $(C_1^{S_1}, g_1^{S_1})$ are consistent with those in $cred_0$.

The authenticity of the signature (i.e., task 2) is verified using the Verify algorithm of SPS-EQ. Tasks 3,4 are combined into an AND-composition proof, proven via NIZK:

$$\begin{aligned} \pi_1 \in & ZKP\{(M, k, d, r, id', d', r', S_1) : \\ & C_0 = h_0^M \cdot h_1^k \cdot h_2^{id} \cdot h_3^{d} \cdot h_4^{r} \\ & \wedge\ C_1^{S_1} = (h_0^M \cdot h_1^k \cdot h_2^{id'} \cdot h_3^{d'} \cdot h_4^{r'})^{S_1} \wedge\ M \geq th\} \end{aligned} \qquad (3)$$

HN can confirm that the UE has a valid $cred_0$ with sufficient balance after verifying the tasks 2-4. However, it remains essential to incorporate a method for double-spending detection and to check if $cred_0$ has already been spent (i.e., the task 1).

Enable Double-spending Detection. When a UE requests to establish a session, the HN first sends a random challenge γ to the UE. The UE then reveals the credential index id within its current $cred_0$ and proves that id matches the third exponent in C_0. Recall that the id is a random number secretly chosen by the UE during the credential issuance (i.e., step ②), and its plaintext was unknown to the HN during issuance. If a $cred$ is used only once (i.e., the HN has not encountered the same id before), id will appear as a dummy value, preserving UE's unlinkability (further explained in Sec V-A2). However, if a misbehaving UE spends it more than once, the HN can recover its identity k through the following method:

During a service request, the UE needs to additionally compute and send $c = k \cdot \gamma + d$ to the HN, and prove it is well-formed using the k and d within $cred_0$. If the HN finds a duplicate id in its database, it retrieves the previous record for the same id, $c' = k \cdot \gamma' + d$. HN can then recover k (i.e., de-anonymize the UE) by solving the following equations involving the same k and d:

$$c = k \cdot \gamma + d, \quad c' = k \cdot \gamma' + d \qquad (4)$$

To ensure that the UE sends the authentic id and computes $c = k \cdot \gamma + d$ in accordance with the secret exponents in C_0, the UE must add an additional statement in the proof π_1:

$$\begin{aligned} \pi_2 \in & ZKP\{(M, k, d, r, id', d', r', S_1) : \\ & C_0 = h_0^M \cdot h_1^k \cdot h_2^{id} \cdot h_3^d \cdot h_4^r \\ & \wedge\ C_1^{S_1} = (h_0^M \cdot h_1^k \cdot h_2^{id'} \cdot h_3^{d'} \cdot h_4^{r'})^{S_1} \\ & \wedge\ M \geq th \quad \wedge\ c = k\gamma + d\} \end{aligned} \tag{5}$$

This guarantees that a malicious UE *cannot* frame others by claiming a forged id, as it lacks the other secrets in $cred_0$ required to prove the id-$cred_0$ binding. Without a valid proof, any falsified id will be rejected.

Summary - Full Procedure Of Session Establishment. (Outlined in the lower portion of Fig. 7.)

$\underline{\mathsf{Spend}(cred_0, (C_1^{S_1}, g_1^{S_1}))}$. At the beginning of the session establishment process, the HN sends a random challenge γ to the UE. The UE computes $c = k\gamma + d$ and the proof π_2, then sends id, c, $cred_0$, and π_2 to the HN.

$\underline{\mathsf{Verify}(cred_0, (C_1^{S_1}, g_1^{S_1}), \pi_2, pk_{HN})}$. HN checks its database for a duplicate id. If a duplicate is found, it de-anonymizes the double-spent UE through the aforementioned method. Otherwise, it validates $cred_0$ (i.e., authenticity of σ_0) using pk_{HN}, verifies π_2, and grants the UE access if all checks pass.

At the end of the session, HN homomorphically subtracts the consumed usage m of the current session from the attribute M in $C_1^{S_1}$ by computing:

$$C_1'^{S_1} = C_1^{S_1} \cdot (h_0^m)^{-1} = (h_0^{(M-m)} \cdot h_1^k \cdot h_2^{id'} \cdot h_3^{d'} \cdot h_4^{r'})^{S_1} \tag{6}$$

This results in a commitment randomized by S_1, which commits to the latest cumulative usage $(M - m)$, the hidden identity k, the two new secret parameters id' and d', and the blinding factor r'. The HN then generates a new SPS-EQ signature $\sigma_1 \leftarrow \mathsf{Sign}(sk_{HN}, (C_1'^{S_1}, g_1^{S_1}))$ and sends them to UE. Upon receiving σ_1 and $(C_1'^{S_1}, g_1^{S_1})$, the UE unblinds $(C_1'^{S_1}, g_1^{S_1})$ to (C_1', g_1), adapts σ_1 to σ_1' using the $\mathsf{Adapt}(\cdot)$ algorithm of SPS-EQ, and obtains a new unlinkable credential $cred_1 : ((C_1', g_1), \sigma_1')$ for future use (same as the $\mathsf{Obtain}(\cdot)$ process in step ③).

We omit the detailed process of the *postpaid* mode as it is simpler and only slightly differs from the *prepaid* mode (as marked in Fig. 7). The only differences are: **(1)** UE does not need to prove the range proof $M \geq th$ in π_2; **(2)** In eq. 6, instead of subtraction, a homomorphic addition operation is used, i.e., compute $C_1'^{S_1} = C_1^{S_1} h_0^m$.

V. Security Analysis and Discussion

A. Security, Anonymity, and Unlinkability

Two factors drive AnonyCall 's anonymity and unlinkability: the frequency of SIP URI reassignment and the computational indistinguishability of the charging procedure.

1) Temporary SIP URI and Unlinkability: The callee discovery functionality inherits anonymity from existing anonymous mobile access schemes [51], [48], [55], [13], [42], such as the security properties of blind signatures or anonymous credentials. Therefore, for the callee discovery phase, we limit our discussion to the unlinkability problem. When an anonymous UE disconnects from the network, its previous IP address becomes invalid. Upon initiating a new registration request, the HN cannot determine if the UE previously owned a particular IP address. By default, SIP URI reassignment typically occurs every 10 to 60 minutes during IMS core registration. To enhance unlinkability, AnonyCall encourages anonymous UEs to refresh their temporary SIP URIs as frequently as possible, ideally every 10 minutes or less. This frequent reassignment can be enforced through operator policies or by allowing UEs to toggle their network service on and off regularly. However, a curious HN may attempt to call or page a victim UE via its temporary SIP URI multiple times within this 10-minute window, potentially identifying linkages among multiple sessions. We acknowledge that there is always a trade-off between privacy and registration frequency. Nonetheless, a 10-minute interval is deemed sufficiently granular to maintain the unlinkability of an anonymous UE.

2) Security and Privacy Analysis of the Charging Method:

Lemma 1. *The $cred$ is unforgeable if the SPS-EQ scheme is* EUF-CMA *secure within the bilinear group model with Type-III pairings, and the zero-knowledge proof meets the* soundness *property.*

Lemma 2. *If the Pedersen Commitment scheme is perfectly hiding, the SPS-EQ scheme is perfectly adaptable, and the ZKP scheme provides* zero-knowledge *property under ROM, then the charging protocol maintains anonymity and unlinkability for a benign UE across different sessions.*

Proofs for Lemmas 1 and 2 are provided in Appendix A-A and A-B, respectively. This section provides an intuitive explanation of these proofs. The charging method offers both unlinkability and anonymity if, for all PPT adversaries $\mathcal{A}$, there exists an efficient simulator Sim such that for all secrets $(M, k, id, d, r, id', d', r', S_0, S_1) \in \mathcal{M}$ with $\phi(M, k, id, d, r, id', d', r', S_0, S_1) = 1$; for all $(\mathsf{BG}) \leftarrow \mathsf{Setup}(1^\lambda)$, $(x_0, \ldots, x_3, h_0, \ldots, h_3, pk_{HN}, sk_{HN}) \leftarrow \mathsf{KeyGen}(\mathsf{BG}, \ell)$; for all $\mathsf{Commit}(\cdot)$ such that $\mathsf{Issue}(\cdot)$ outputs 1; and for all $\mathsf{Spend}(\cdot)$ such that $\mathsf{Verify}(\cdot)$ outputs 1. That is to say, an adversarial HN $\mathcal{A}$'s view, given the proof, can be simulated by Sim given only ϕ, the valid signing key sk_{HN}, and the secret trapdoors $x_0, \ldots, x_3$ corresponding to the public verification parameters $g_1, h_0, \ldots, h_3$. If the secrets $\mathcal{M}$ embedded within a $cred$ (including its successor $cred'$) are hidden across different network accesses and appear uniformly randomly distributed, the protocol meets the criteria of *unlinkability*. These properties can be proved within the same game. The difficulty of de-anonymizing or distinguishing linkage among any two anonymous sessions is reduced to the hardness of DLP and the perfect adaptability of the SPS-EQ scheme. Essentially, the proofs π_0 and π_2 appear random and only reveal the validity of the statements, while an adapted SPS-EQ signature σ' cannot link back to its original form σ that HN has seen.

Identity-hiding. The $cred$ enables anonymous and unlinkable charging for a UE by its HN. The same k value in different $cred$ instances is effectively concealed, and any two sets of attributes (M, k) and (M', k') in successive $cred$ and $cred'$ instances are unlinkable. These properties are ensured by the *hiding* property of the Pedersen commitment, the *adaptability* of SPS-EQ, and the *zero-knowledge* property (i.e., *witness indistinguishability*) of ZKPs under the ROM, through the use of uniformly random values, such as r, S_0, id, and parameters from the NIZK process (Appendix A-B). Consequently, AnonyCall ensures that the identity of a benign UE (i.e., k) and the actual balance M remain undisclosed to its HN. Therefore, as long as there is no double-spending, AnonyCall guarantees that different session establishments, whether made by the same UE, cannot be correlated or identified, rendering it impossible for an HN to trace a particular UE.

Uniqueness of id and Collision Prevention. Since every UE needs to obtain a new $cred$ at the beginning of each month, the HN can reset all the stored id that it has seen from its database by the end of each month. While two users may, in rare cases, choose the same secret id during the $cred$ issuance, the possibility is negligible. For example, a 256-bit prime p ($id \in \mathbb{Z}_p$) provides 2^{256} possible values for id, therefore, the message space would provide more than enough unique possibilities to allow each UE of the same HN to carry a unique id inside each month. We can further quantify this using the birthday paradox: assuming 1 million users, the probability of two users selecting the same secret id is: $\Pr[\text{id\_collision}] \approx \frac{1}{2} \cdot \frac{10^{12}}{2^{256}} \approx 2^{-218}$. That said, we can incorporate a lightweight safeguard during the initial credential issuance to prevent id collision: the user can additionally provide h_2^{id}, where id remains secret due to the hardness of the discrete logarithm problem. The HN can maintain a temporary list of all issued h_2^{id} values for the current issuance period (e.g., per month). If a newly requested value h_2^{id} matches an existing one, the MNO can prompt the user to retry with a different set of secret parameters. This approach can ensure strong collision resistance without revealing id, S_i or any other secrets associated with $cred$. It also does not introduce linkability, as the hardness of the Computational Diffie-Hellman (CDH) problem guarantees that, even given h_2, h_2^{id} and $h_2^{id \cdot S_i}$, an adversary cannot infer either id or S_i, nor determine whether two components share the same id.

B. Interoperability, Compatibility, and Practicality

Q1: How to dial a SIP URI instead of a Phone number? In cellular networks, subscribers often do not know their SIP URI by default. The IMS core translates both the caller's and callee's phone numbers into SIP URIs at the start of session establishment. Some providers support direct SIP URI dialing (e.g., Verizon, AT&T). On the UE side, most mobile devices support placing and receiving calls using registered SIP URIs through either native dialers or third-party apps (e.g., Linphone, Zoiper, Bria, Zoom). For instance, our experiments utilized the Linphone app (Fig. 9, Appendix).

Q2: If a caller can obtain the callee's SIP URI out-of-band, why not communicate directly through this channel? Callees may not always have Internet access or the ability to use an app. Phone calls should still be received regardless of Internet connectivity, for which the native voice service is still the de facto choice. On the other hand, the out-of-band authentication channel is between the caller and the callee's authentication agent, not between the caller and the callee directly. The channel is also not required to provide the bandwidth requirement as with a voice call.

Q3: Will all MNOs need to make modifications in order to make AnonyCall work, and what incentives do they have to accept and deploy AnonyCall? Using a SIP URI is a common method for making phone calls. For anonymous callee discovery, the only change for an MNO is the SIP URI assignment during IMS core registration. Instead of using the semi-permanent SIP URI of the UE directly, the HN assigns a temporary SIP URI based on the UE's current IMS IP address, which becomes invalid when the IP address is no longer in use. This is feasible since a subscriber can have multiple SIP URIs simultaneously in current mobile networks. Thus no modifications are needed for existing call routing protocols, e.g., SIP signaling. For the charging method, because no new entities are created, the MNO only requires algorithm-level changes within the relevant network functions, such as CHF.

Furthermore, AnonyCall *only requires the anonymous UE's HN to deploy the changes for its two functionalities, and it does not require all MNOs to implement its functionalities to make it work.* Since in a mobile network, a roaming UE's SIP URI and the IP address for IMS are always managed by its HN, which charges the UE based on observed usage. Therefore, AnonyCall remains functional even when an anonymous UE roams to a foreign network that does not deploy it. A non-anonymous caller A from a conventional operator can always reach an anonymous callee B, since B's home IMS can resolve the current IMS IP address associated with B from B's temporary SIP URI, inform A's home IMS, and establish the call session through standard inter-IMS procedures.

We do not anticipate that all MNOs will adopt these methods to enhance subscriber privacy. Instead, AnonyCall can be offered by a dedicated MVNO specializing in private and anonymous services for privacy-conscious mobile users.

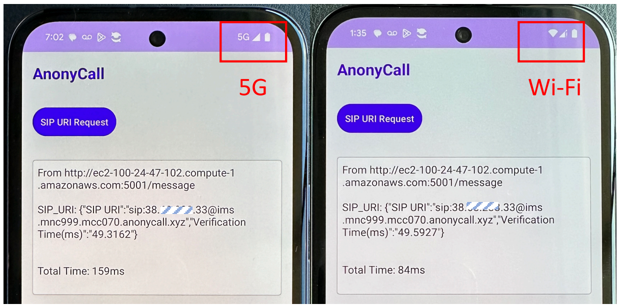
To ensure the *confidentiality of the temporary SIP URI* from the service provider (e.g., an MVNO), a client-side encryption model similar to those used by popular password manager applications on iOS and Android devices can be adopted. In this model, the SIP URI is encrypted locally on the device using a device-resident cryptographic key, and only the resulting ciphertext is synchronized with the cloud agent. When sharing with an authorized caller is required, the device decrypts the SIP URI and re-encrypts it under the caller's public key, ensuring that the cloud agent remains unaware of the plaintext. Alternatively, other common cloud data protection mechanisms such as Attribute-Based Encryption (ABE) or Proxy Re-Encryption (PRE) can be employed to achieve confidentiality and flexible access control of the SIP URI.

Q4: Why not just use a Tor-like network for anonymous communication? Anonymous communication methods like MixNets and Tor are designed to route untraceable data packets over the conventional Internet, without centralized service providers managing infrastructure or usage-based billing. In contrast, phone calls rely on cellular IP networks, which provide QoS guarantees and require billing mechanisms.

Q5: What if a malicious HN intercepts the call session? (Potentials to support end-to-end encryption (E2EE)) Due to space limits, we defer this question to the Appendix A-C.

VI. Implementation and Evaluation

Setup and Environment. (1) MNO. To model the voice call in mobile networks, we deploy a SIP signaling server by using a standard VoLTE/VoNR IMS core (`Kamailio` [6]), which includes the essential IMS functions such as P-CSCF, S-CSCF, and I-CSCF, and connects to the 5GC based on the standard



From http://ec2-100-24-47-102.compute-1 .amazonaws.com:5001/message

SIP_URI: {"SIP URI":"sip:38.[redacted].33@ims .mnc999.mcc070.anonycall.xyz","Verification Time(ms)":"49.3162"}

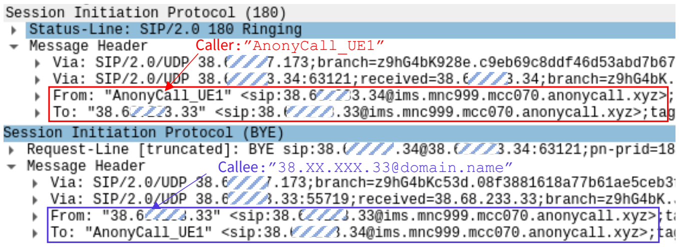
Total Time: 159ms

From http://ec2-100-24-47-102.compute-1 .amazonaws.com:5001/message

SIP_URI: {"SIP URI":"sip:38.[redacted].33@ims .mnc999.mcc070.anonycall.xyz","Verification Time(ms)":"49.5927"}

Total Time: 84ms

Fig. 5. Caller side - successfully authenticated by the callee and obtained the callee's SIP URI. (*RTT under 5G and Wi-Fi)



```
Session Initiation Protocol (180)
 ▸ Status-Line: SIP/2.0 180 Ringing
 ▾ Message Header
    ▸ Via: SIP/2.0/UDP 38.[redacted]7.173;branch=z9hG4bK928e.c9eb69c8ddf46d53abd7b67
    ▸ Via: SIP/2.0/UDP 38.[redacted]3.34:63121;received=38.[redacted]3.34;branch=z9hG4bK.
    ▸ From: "AnonyCall_UE1" <sip:38.6[redacted]3.34@ims.mnc999.mcc070.anonycall.xyz>;
    ▸ To: "38.6[redacted]3.33" <sip:38.[redacted].33@ims.mnc999.mcc070.anonycall.xyz>;tag
Session Initiation Protocol (BYE)
 ▸ Request-Line [truncated]: BYE sip:38.[redacted].34@38.[redacted]3.34:63121;pn-prid=18
 ▾ Message Header
    ▸ Via: SIP/2.0/UDP 38.6[redacted].173;branch=z9hG4bKc53d.08f3881618a77b61ae5ceb3
    ▸ Via: SIP/2.0/UDP 38.6[redacted].33:55719;received=38.68.233.33;branch=z9hG4bK..
    ▸ From: "38.6[redacted].33" <sip:38.[redacted].33@ims.mnc999.mcc070.anonycall.xyz>;ta
    ▸ To: "AnonyCall_UE1" <sip:38.[redacted].34@ims.mnc999.mcc070.anonycall.xyz>;ta
```

Fig. 6. SIP Messages during Anonymous Call Establishment
* SIP Packets `180 Ringing` and `BYE` (captured by *Wireshark*)
* IMS domain name: `ims.mnc999.mcc070.anonycall.xyz`

open source project `Open5GS` [7]. The 5GC and IMS core are deployed on a desktop running Ubuntu 20.04, with an Intel Core i7-11700k, 3.6GHZ, 8-core, 64-bit CPU. **(2) UE.** The applications on the UE side are implemented on a `Google Pixel 8a` (8 GB RAM, Google Tensor G3, Titan M2 security co-processor). The VoIP application `Linphone` [9] is used to dial the temporary SIP URI on both an `iPhone 13 Pro Max` and a `Google Pixel 8a`. **(3) Authentication Agent.** The cloud-based authentication agent is deployed on an AWS Linux OS (AWS EC2, t2.micro instance, east US).

A. *Anonymous Callee Discovery*

The evaluation for this functionality involves two parts: (1) an *out-of-band authentication framework*, the main building blocks are the authentication agent functions (phone-based and cloud-based). (2) the *anonymous call establishment* process between two phones through a standard IMS system.

Authentication Agent. We deployed a prototype Android application to evaluate out-of-band caller authentication in two policy modes. **a. Simple Mode:** This mode utilizes PKI-based digital certificates and is useful when the caller and callee already know each other or when applying simple policies, such as verifying if the caller is a legitimate phone user rather than a spammer using an automatic dialer. **b. Flexible Mode:** This mode is based on DPKI and standardized VCs. It supports customized authentication policies and establishes trust between parties with no prior knowledge of each other.

To test the *simple* mode, we installed a standard X.509 digital certificate provided by the OpenSSL library on the caller's side. This enables the callee to authenticate the caller by verifying its digital certificate, ensuring essential caller authentication functionalities, and securing communications during the temporary SIP URI sharing process. Additionally, we deployed the callee's authentication agent on an AWS server. As shown in Fig. 5, we tested the complete process from the caller sending its request to the agent, bypassing authentication, and obtaining the temporary SIP URI. The entire process takes between 50 and 180 milliseconds, with the certificate utilizing RSA-2048 bit keys for encryption and the SHA256 algorithm for hashing. Since certificate verification is a lightweight process on modern processors (e.g., only a few milliseconds with RSA-2048), the overhead of this authentication process is influenced more by network conditions than by cryptographic computations. To test the *flexible* mode, we used the DPKI offered by the `ION` platform [4], which is part of the W3C DID standard-compliant Microsoft `Entra` project [10]. In addition, we used a demo onboarding platform provided by `Entra` to acquire the VCs for the digital wallet on UE. The caller can contact the callee's authentication agent and generate a presentation from its VC according to the callee's policy. Our test used the default VC provided by `Entra`, which is a digital driver's license containing eight attributes. The Fig. 8 of the Appendix shows an example of a UE obtaining the VC from `Entra`, reaching the callee's authentication agent, and deriving a presentation through VC for authentication. The verification (i.e., presentation) time for all attributes within VC in one shot takes around 650 to 780 milliseconds. The main latency is due to the cryptographic computation of VC and the communication latency between the cloud server and the UE.

Nevertheless, as discussed previously, a caller does not need to authenticate itself to a callee for each phone call. Once a caller has been authenticated, the authentication agent can provide them access rights to the SIP URI for a longer duration, e.g., one year. The caller can retrieve the callee's most recent SIP URI directly from the agent, eliminating the need for costly VC presentation before each call session. The overhead of the above process is then confined to network conditions and excludes cryptographic authentication. For example, our prototype UE takes ∼80 milliseconds to retrieve a SIP URI from the AWS-based agent under Wi-Fi settings.

Call Establishment. On the MNO side, we manually registered two prototype UEs with our IMS core, one anonymous (`38.XX.XXX.33@domain.name`) and one in standard mode (`AnonyCall_UE1@domain.name`), and mapped each UE's current SIP URI to an IP address in the subscriber database. After successful out-of-band authentication, the caller (`AnonyCall_UE1`) can directly dial the anonymous callee's SIP URI without relying on protocol-level modifications, thus introducing no overhead. The two UEs can successfully talk to each other, with the call data packets routed through UDP, as RTP runs over UDP in VoLTE/VoNR. We also captured and examined the SIP messages between them throughout the call session. Two SIP message examples are shown in Fig. 6, where all SIP message bodies contain only the temporary SIP URI, revealing no permanent information about the two anonymous UEs.

B. *Charging Method*

For the charging protocol, we assessed the computational overhead of the cryptographic protocols on both the UE and HN (desktop) using the cryptographic libraries `Cryptimeleon` [18] and `mcl` [5]. Our experiment employs the `bn254` curve to generate the bilinear group, chosen for

TABLE II. **TIME CONSUMPTION** (IN MILLISECONDS) OF DIFFERENT STAGES IN THE CHARGING PROTOCOL (*AVERAGE OF 100 RUNS)

	Setup			Postpaid	Prepaid
UE	Commit + Obtain	8.01	Spend	26.72	27.35
HN	Issue	4.65	Verify	47.88	49.14

its efficiency in computing pairing operations [39]. It operates over a 254-bit field and offers a security level of $\lambda = 128$ bits, i.e., the order p of the curve is a 254-bit prime, allowing the message space for the balance attribute M to be up to approximately $2^{254} - 1$. This large message space is sufficient to represent the balance attribute M in fine-grained units, such as the total number of milliseconds in a month, thus enabling precise billing capabilities. In our experiment, M is implemented using Java `Bigintegers`. The time consumption of each stage (or algorithm) in the charging protocol is listed in Table II. On the HN side, a significant computational expense is incurred during the verification of the SPS-EQ signature within $cred_0$, which requires computing four pairing operations. Additionally, the prepaid mode is more costly than the postpaid mode since the HN needs to verify an additional range proof to check if the UE has sufficient balance. Our evaluation implemented the classic range proof [21], but its efficiency can be further optimized using more advanced range proofs, such as Bulletproofs [20].

On the UE side, the computations for the successor credential and the first three statements in π_2 (i.e., π_1) under the Fiat-Shamir heuristic NIZK method do not depend on the fresh challenge γ sent from the HN. Therefore, these can be pre-computed by UE to expedite the call establishment process. Specifically, at the end of each session, after the HN issues a new $cred$ with the updated balance, the UE can store it and immediately prepare for its successor credential and π_1. Consequently, these steps will not introduce latency to the call establishment time. The only computation required at the beginning of the session is the simple response including the HN's fresh challenge ($c = k\gamma + d$), used for double-spending detection, and the proofs of its correctness (explained in Appendix A-B2). Thus, we can further minimize the introduced latency by having the UE compute π_1 offline.

Summary - The overall introduced latency to the call establishment process by ANONYCALL. The out-of-band callee authentication scheme can be performed once and remain valid for an extended period. Additionally, the latency introduced by the anonymous charging method can be significantly minimized by having the UE pre-compute most computations in advance. Thus, the estimated latency introduced to the call establishment process by ANONYCALL can be well below 200 milliseconds. This estimate includes the time for the caller to retrieve the latest SIP URI from the cloud-based authentication agent ($\sim$ 80 ms) and our measurements on the total time of spending and verifying $cred$ in the pre-paid mode without pre-computations ($\sim$ 76 ms).

VII. RELATED WORK

Anonymous Peer Discovery and Communication. Most existing anonymous peer discovery and communication applications and research that can protect users' identity and communication content privacy lie in building an ad-hoc network that does not rely on a pre-existing infrastructure or any centralized management. Examples include COVID-19 contact tracing, Apple's FindMy network, and various research projects focused on creating ad-hoc privacy-preserving networks over BLE, such as [41], [53], [45]. However, such mechanisms can not be applied to the calling system, as call services must rely on a cellular or non-cellular communications operator to operate the infrastructure and route the data packets between caller and callee. [14] proposed a private session establishment scheme for protecting communication content confidentiality from untrusted MNOs while allowing authorized parties to perform lawful interception on the communication content. However, their method requires the involved callers and callees are not anonymous.

Caller Authentication. The out-of-band caller authentication mechanism for ANONYCALL's anonymous callee discovery shares some similarities with prior methods. For example, AuthLoop [47], Authenticall [46], and the STIR/SHAKEN framework [30] implemented by US carriers provide caller ID authentication to combat caller ID spoofing and robocalls. However, these methods do not protect users' identity privacy from untrusted MNOs. UCBlocker [29] is a whitelist-based call-blocking application that relies on attribute-based credentials to authenticate callers. In contrast, our out-of-band caller authentication serves a different purpose. It aims to provide a secure channel to pass the anonymous callee's SIP URI to a valid caller, enabling the caller to initiate phone calls within the cellular network rather than focusing on call blocking.

Usage-based Charging. Traditional e-cash schemes [24], [25], [15], [23] enable anonymous payments but only support single-use cases, lacking fine-grained charging and are suitable only for prepaid applications. More flexible systems, such as those in [40], [36], enable cumulative usage through homomorphic commitments and ZKP, but do not support partial spending and require users to explicitly reveal their balance, causing linkability issues. [37], [17] addressed this by incorporating blind signature schemes like CL-signatures [16] and PS-signatures [44], allowing users to hide their exact balance from the verifier. Building on [17], the scheme proposed in [19] eliminates the need for ZKPs during spending by leveraging SPS-EQ adaptability. Additionally, their scheme addresses backward unlinkability, preserving the anonymity of a double-spending user's prior transactions in multi-verifier applications. In contrast, ANONYCALL targets a single-verifier service model that prioritizes time sensitivity and low computational overhead during call setup. It intentionally supports conditional unlinkability, as protecting the anonymity of a double-spending UE's previous sessions is unnecessary.

VIII. CONCLUSIONS

ANONYCALL is a privacy-preserving call management system that supports native phone calls and user charging while enabling anonymous mobile network access. It is interoperable and compatible with standard cellular networks. Our evaluation shows that it introduces acceptable latency to call setup, paving the way for private and functional mobile communication.

ACKNOWLEDGMENT

This work was supported in part by the US National Science Foundation under grants 2247560, 2247561, 2247562, 2154929, and 2312447, by the Office of Naval Research under grant N00014-24-1-2730, and by the Virginia Commonwealth Cyber Initiative (CCI).

References

[1] https://www.digitalidentity.gov.au/tdif.

[2] DID Specification Registries. https://www.w3.org/TR/did/upcoming/.

[3] eIDAS Regulation — digital-strategy.ec.europa.eu. https://digital-strategy.ec.europa.eu/en/policies/eidas-regulation.

[4] GitHub - decentralized-identity/ion. https://github.com/decentralized-identity/ion.

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

[6] GitHub - kamailio/kamailio: Kamailio - The Open Source SIP Server for large VoIP and real-time communication platforms. https://github.com/kamailio/kamailio.

[7] GitHub - open5gs/open5gs: Open5GS is a C-language Open Source implementation for 5G Core and EPC, i.e. the core network of LTE/NR network (Release-17). https://github.com/open5gs/open5gs.

[8] iCloud data security overview - Apple Support — support.apple.com. https://support.apple.com/en-us/102651.

[9] Linphone open source VoIP SIP softphone - voice, video and instant messaging. https://www.linphone.org/.

[10] Microsoft Entra Verified ID — Microsoft Security — microsoft.com. https://www.microsoft.com/en-us/security/business/identity-access/microsoft-entra-verified-id.

[11] Trust Framework. https://diacc.ca/trust-framework/.

[12] Verifiable Credentials Data Model v2.0 — w3.org. https://www.w3.org/TR/vc-data-model-2.0/.

[13] Rabiah Alnashwan, Yang Yang, Yilu Dong, Prosanta Gope, Behzad Abdolmaleki, and Syed Rafiul Hussain. Strong privacy-preserving universally composable aka protocol with seamless handover support for mobile virtual network operator. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 2057–2071, 2024.

[14] Ghada Arfaoui, Olivier Blazy, Xavier Bultel, Pierre-Alain Fouque, Thibaut Jacques, Adina Nedelcu, and Cristina Onete. How to (legally) keep secrets from mobile operators. In *Computer Security–ESORICS 2021: 26th European Symposium on Research in Computer Security, Darmstadt, Germany, October 4–8, 2021, Proceedings, Part I 26*, 2021.

[15] Foteini Baldimtsi, Melissa Chase, Georg Fuchsbauer, and Markulf Kohlweiss. Anonymous transferable e-cash. In *IACR International Workshop on Public Key Cryptography*, 2015.

[16] Foteini Baldimtsi and Anna Lysyanskaya. Anonymous credentials light. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013.

[17] Johannes Blömer, Jan Bobolz, Denis Diemert, and Fabian Eidens. Updatable anonymous credentials and applications to incentive systems. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019.

[18] Jan Bobolz, Fabian Eidens, Raphael Heitjohann, and Jeremy Fell. Cryptimeleon: A library for fast prototyping of privacy-preserving cryptographic schemes. *IACR Cryptol. ePrint Arch.*, 2021.

[19] Jan Bobolz, Fabian Eidens, Stephan Krenn, Daniel Slamanig, and Christoph Striecks. Privacy-preserving incentive systems with highly efficient point-collection. In *Proceedings of the 15th ACM Asia Conference on Computer and Communications Security*, 2020.

[20] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE symposium on security and privacy (SP)*, 2018.

[21] Jan Camenisch, Rafik Chaabouni, and Abhi Shelat. Efficient protocols for set membership and range proofs. In *Advances in Cryptology - ASIACRYPT 2008, 14th International Conference on the Theory and Application of Cryptology and Information Security*, 2008.

[22] Jan Camenisch and Markus Stadler. Efficient group signature schemes for large groups. In *Annual International Cryptology Conference*, 1997.

[23] Sébastien Canard and Aline Gouget. Divisible e-cash systems can be truly anonymous. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2007.

[24] David Chaum. Blind signatures for untraceable payments. In *Advances in Cryptology: Proceedings of Crypto 82*, 1983.

[25] David Chaum and Torben Pryds Pedersen. Transferred cash grows in size. In *Workshop on the Theory and Application of of Cryptographic Techniques*, 1992.

[26] European Commission. Eu digital identity wallet (eudi wallet) large-scale pilots. https://digital-strategy.ec.europa.eu/en/policies/eudi-wallet-implementation, 2025.

[27] Aisling Connolly, Pascal Lafourcade, and Octavio Perez Kempner. Improved constructions of anonymous credentials from structure-preserving signatures on equivalence classes. In *IACR International Conference on Public-Key Cryptography*, 2022.

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

[29] 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)*, 2023.

[30] FCC. Mandating stir/shaken to combat spoofed robocalls. https://www.fcc.gov/document/mandating-stirshaken-combat-spoofed-robocalls-0, 2020.

[31] Federal Communications Commission (FCC). FCC Fines AT&T, Sprint, T-Mobile, and Verizon Nearly $200 Million for Illegally Sharing Access to Customers' Location Data. https://docs.fcc.gov/public/attachments/DOC-402213A1.pdf, 2024.

[32] 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*, 1986.

[33] Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Practical round-optimal blind signatures in the standard model. In *Annual Cryptology Conference*, 2015.

[34] Georg Fuchsbauer, Christian Hanser, and Daniel Slamanig. Structure-preserving signatures on equivalence classes and constant-size anonymous credentials. *Journal of Cryptology*, 2019.

[35] Lucjan Hanzlik and Daniel Slamanig. With a little help from my friends: Constructing practical anonymous credentials. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021.

[36] Gunnar Hartung, Max Hoffmann, Matthias Nagel, and Andy Rupp. Bba+ improving the security and applicability of privacy-preserving point collection. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[37] Max Hoffmann, Michael Klooß, Markus Raiber, and Andy Rupp. Black-box wallets: Fast anonymous two-way payments for constrained devices. *Proceedings on privacy enhancing technologies*, 2020.

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

[39] IETF. Pairing-Friendly Curves. https://www.ietf.org/archive/id/draft-irtf-cfrg-pairing-friendly-curves-02.html.

[40] Tibor Jager and Andy Rupp. Black-box accumulation: Collecting incentives in a privacy-preserving way. *Proceedings on Privacy Enhancing Technologies*, 2016.

[41] Matthew Lentz, Viktor Erdélyi, Paarijaat Aditya, Elaine Shi, Peter Druschel, and Bobby Bhattacharjee. {SDDR}:{Light-Weight}, secure mobile encounters. In *23rd USENIX Security Symposium (USENIX Security 14)*, 2014.

[42] Zhihong Luo, Silvery Fu, Natacha Crooks, Shaddi Hasan, Christian Maciocco, Sylvia Ratnasamy, and Scott Shenker. {LOCA}: A {Location-Oblivious} cellular architecture. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*, pages 1621–1646, 2023.

[43] Torben Pryds Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In *Annual international cryptology conference*, 1991.

[44] David Pointcheval and Olivier Sanders. Short randomizable signatures. Cryptology ePrint Archive, Paper 2015/525, 2015.

[45] Amogh Pradeep, Hira Javaid, Ryan Williams, Antoine Rault, David Choffnes, Stevens Le Blond, and Bryan Alexander Ford. Moby: A blackout-resistant anonymity network for mobile devices. *Proceedings on Privacy Enhancing Technologies*, 2022.

[46] Bradley Reaves, Logan Blue, Hadi Abdullah, Luis Vargas, Patrick Traynor, and Thomas Shrimpton. {AuthentiCall}: Efficient identity and content authentication for phone calls. In *26th USENIX Security Symposium (USENIX Security 17)*, 2017.

[47] Bradley Reaves, Logan Blue, and Patrick Traynor. {AuthLoop}:{End-to-End} cryptographic authentication for telephony over voice channels. In *25th USENIX Security Symposium (USENIX Security 16)*, 2016.

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

[49] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In *Conference on the Theory and Application of Cryptology*, 1989.

[50] 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*, 2012.

[51] 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*, 2020.

[52] Linux Foundation Decentralized Trust. Credebl: Open-source decentralized identity verifiable credentials platform. https://www.lfdecentralizedtrust.org/projects/credebl, 2025.

[53] Jean-Luc Watson, Tess Despres, Alvin Tan, Shishir G Patil, Prabal Dutta, and Raluca Ada Popa. Nebula: A privacy-first platform for data backhaul. In *2024 IEEE Symposium on Security and Privacy (SP)*, 2024.

[54] Yang Yang, Quan Shi, Prosanta Gope, Behzad Abdolmaleki, and Biplab Sikdar. Pgus: Pretty good user security for thick mvnos with a novel sanitizable blind signature. In *2025 IEEE Symposium on Security and Privacy (SP)*, pages 1140–1158. IEEE, 2025.

[55] Hexuan Yu, Changlai Du, Yang Xiao, Angelos Keromytis, Chonggang Wang, Robert Gazda, Y Thomas Hou, and Wenjing Lou. AAKA: An anti-tracking cellular authentication scheme leveraging anonymous credentials. In *Network and Distributed System Security Symposium (NDSS)*, 2024.

Appendix A
Preliminaries and Security Analysis

A. SPS-EQ

This section provides the essential background of the SPS-EQ signature scheme. The formal definitions follow [34], [33].

1) Bilinear Map and Bilinearity: In SPS-EQ, bilinear pairings are used to construct and verify signatures that preserve the algebraic structure of the underlying groups. A bilinear group generator BG outputs $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$. $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ are cyclic groups of prime order p. $P \in \mathbb{G}_1$, $\hat{P} \in \mathbb{G}_2$, i.e., P and $\hat{P}$ are two generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. There exists a bilinear mapping or pairing function such that $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$. For all $P \in \mathbb{G}_1, \hat{P} \in \mathbb{G}_2$, and $a, b \in \mathbb{Z}_p$, $e(aP, b\hat{P}) = e(P, \hat{P})^{ab}$. SPS-EQ operates under the Type-III pairing settings such that there is no efficiently computable isomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$.

2) Equivalence Classes: Consider a vector of messages $(M_i)_{i\in[\ell]}$ from the group $(\mathbb{G}^*)^\ell$. We define an equivalence relation $\mathcal{R}$ on $(\mathbb{G}^*)^\ell$ such that two vectors $\vec{M}$ and $\vec{M}'$ belong to the same equivalence class if and only if there exists a scalar $\mu \in \mathbb{Z}_p^*$ for which the following relation $\mathcal{R}$ holds:

$$\{(\vec{M}, \vec{M}') \in (\mathbb{G}^*)^\ell \times (\mathbb{G}^*)^\ell \mid \exists \mu \in \mathbb{Z}_p^* : \vec{M}' = \mu\vec{M}\} \subseteq (\mathbb{G}^*)^{2\ell}$$

This means that $\vec{M}$ and $\vec{M}'$ are in the same equivalence class under $\mathcal{R}$ if $\vec{M}'$ can be derived by scaling each component of $\vec{M}$ by the scalar μ. Note that, in the context of elliptic curves, scalar multiplication (additive notation) is analogous to exponentiation in finite fields (multiplicative notation). Here, we use additive notation to introduce the SPS-EQ scheme for ease of understanding, as points on elliptic curves are naturally added together. However, in Sec IV, we adopted multiplicative notation, such as $\vec{M}' = M^\mu$, instead of the additive form $\vec{M}' = \mu \cdot M$. This choice ensures consistency with Pedersen commitment, which is naturally expressed multiplicatively due to its use of exponentiation, making calculations and proofs of our charging scheme more intuitive. Despite the difference in notation, both additive and multiplicative forms represent the same underlying bilinear properties of the pairing function in SPS-EQ. The choice of notation only affects how group operations are expressed, not the fundamental mathematical structure, functionality, or security of SPS-EQ.

Algorithms of SPS-EQ. The algorithm KeyGen only executes once during the system setup (by MNO). The algorithm Adapt can be seen as a function that validates the matching relation between the message and its signature under the blinding factor μ, i.e., $\mathsf{Sign}(sk_{HN}, \mu\vec{M}; \psi y)$, and can be verified by running $\mathsf{Verify}(pk_{HN}, \vec{M}', \sigma')$.

$\underline{\mathsf{KeyGen}(\mathsf{BG}, 1^\ell)}$. On input a bilinear-group generator BG, and the message vector length $\ell > 1$, output $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, \hat{P})$. Choose $(x_i)_{i\in[\ell]} \xleftarrow{R} (\mathbb{Z}_p^*)^\ell$, set secret key $sk_{HN} = (x_i)_{i\in[\ell]}$, compute public key $pk_{HN} = (\hat{X}_i)_{i\in[\ell]} = (x_i\hat{P})_{i\in[\ell]}$ and output the secret-public key pair (sk_{HN}, pk_{HN}) for the home network (i.e., signer).
$\underline{\mathsf{Sign}(sk_{HN}, \vec{M})}$. On input a representative message vector $\vec{M} = (M_i)_{i\in[\ell]}$ of equivalence class $[\vec{M}]_\mathcal{R}$ and a secret key $sk_{HN} = (x_i)_{i\in[\ell]} \in (\mathbb{Z}_p^*)^\ell$, return $\perp$ if $M_i \notin \mathbb{G}_1^*$ for some $i \in [\ell]$. Else, choose $y \xleftarrow{R} \mathbb{Z}_p^*$ and output the signature $\sigma = (Z, Y, \hat{Y})$, in which

$$Z = y \sum_{i\in[\ell]} x_i M_i, \quad Y = \frac{1}{y}P, \quad \hat{Y} = \frac{1}{y}P.$$

$\underline{\mathsf{Verify}(\vec{M}, \sigma, pk_{HN})}$. On input a representative message vector $\vec{M} = (M_i)_{i\in[\ell]}$ of equivalence class $[\vec{M}]_\mathcal{R}$ and a signature $\sigma = (Z, Y, \hat{Y})$ and public key $pk_{HN} = (\hat{X}_i)_{i\in[\ell]}$, output 0 if for some $i \in [\ell]$:
$M_i \notin \mathbb{G}_1^*$ or $\hat{X}_i \notin \mathbb{G}_2^*$ or if $Z \notin \mathbb{G}_1$ or $Y \notin \mathbb{G}_1^*$ or $\hat{Y} \notin \mathbb{G}_2^*$.
Return 1 if the following equations hold and 0 otherwise:

$$\prod_{i\in[\ell]} e(M_i, \hat{X}_i) = e(Z, Y) \quad \wedge \quad e(Y, \hat{P}) = e(P, \hat{Y}).$$

$\underline{\mathsf{Adapt}(\vec{M}, \sigma, \mu, pk_{HN})}$. on input a representative $\vec{M} = (M_i)_{i\in[\ell]}$ of equivalence class $[\vec{M}]_\mathcal{R}$, the orginial signature $\sigma = (Z, Y, \hat{Y})$ and public key pk_{HN}, return $\perp$ if $\mathsf{Verify}(\vec{M}, \sigma, pk_{HN}) = 0$. Otherwise it computes $\psi \xleftarrow{R} \mathbb{Z}_p^*$

and output an adapted signature σ' on a new representative $\vec{M}' = \mu \cdot \vec{M}$ as: $\sigma' = \left(\psi\mu Z, \frac{1}{\psi}Y, \frac{1}{\psi}\hat{Y}\right)$

$$= \left(\psi y \sum_{i\in[\ell]} x_i \mu M_i, \frac{1}{\psi}\frac{1}{y}P, \frac{1}{\psi}\frac{1}{y}\hat{P}\right)$$

Unforgeability. The unforgeability of $cred$ can be reduced to proving two key points: **(1)** the unforgeability of an SPS-EQ signature-message pair, since a $cred$ consists of a message and an SPS-EQ signature, e.g., $cred_0 : ((C_0, g_1), \sigma_0)$. The SPS-EQ scheme has been proven to be EUF-CMA secure under Type-III pairing as shown in [34], [33]. **(2)** Additionally, it relies on the *soundness* property of the ZKP, meaning that a dishonest UE cannot convince the HN that it knows the secrets embedded in a $cred$, which is further explained in the next section.

B. Proofs of π_0, π_1, π_2 and Security Proofs

In π_0, the UE must prove knowledge of the secret values $(M_0, k_0, id,\ d, r, S_0)$ such that for $C^{S_0} = (h_2^{id} \cdot h_3^d \cdot h_4^r)^{S_0}$, $M_0 = k_0 = 0$. Essentially, this means demonstrating that the exponents for the fixed public values h_0 and h_1 are zero. Recall that C^{S_0} and $h_0, \ldots, h_4$ are known to HN, but C is unknown to HN. To simplify notation, let $A = C^{S_0}$ and denote the exponents for h_2, h_3, h_4 as u, v, w, respectively. Thus, the proof process can be expressed as $A = h_2^u \cdot h_3^v \cdot h_4^w$. Any NP-relation has a zero-knowledge proof of knowledge, so the randomness id, d, r and S_0 chosen by the UE have no bearing on the proving goal. The UE can adapt the randomness to u, v, w, as long as it can prove that the exponents for h_0 and h_1 are zero. This proving process is similar to Okamoto's protocol. We apply the Fiat-Shamir heuristic to transform all the proofs involved in the charging process into NIZK:

1. Commit. UE selects random values $r_u,\ r_v,\ r_w \in \mathbb{Z}_p$ and computes $a = h_2^{r_u} \cdot h_3^{r_v} \cdot h_4^{r_w}$

2. Generate the challenge and response using Fiat-Shamir. UE computes the challenge $e = H(A||a)$ as a hash of a and A, then computes:

$$z_u = r_u + e \cdot u, \quad z_v = r_v + e \cdot v, \quad z_w = r_w + e \cdot w (\mathrm{mod} p)$$

UE sends (a, z_u, z_v, z_w) to the HN.

3. Verification. HN computes the challenge independently:

$$e' = H(A||a)$$

The HN should get the same value for e' as the UE did for e if the a and public value A are correct. HN checks the equation:

$$A^{e'} \cdot a \overset{?}{=} h_2^{z_u} \cdot h_3^{z_v} \cdot h_4^{z_w}$$

This construction ensures that the HN is convinced that the UE knows the values u, v, and w without revealing them, while verifying that $A = h_2^u \cdot h_3^v \cdot h_4^w$, which implicitly proves that the exponents for h_0, h_1 are zero.

1) Proof Sketch of Lemma 1,2: **Witness Indistinguishability.** In the proof above, the UE's witness (i.e., secret) is (u, v, w). The proving process ensures *witness indistinguishability*, as the message (a, z_u, z_v, z_w) (sent in step **2**) observed by the potentially dishonest HN $\mathcal{A}$ is statistically independent of the secret (u, v, w). To demonstrate this, consider another witness (u', v', w'). There are unique values $r'_u, r'_v, r'_w \in \mathbb{Z}_p$ that can produce the same message (a, z_u, z_v, z_w) for a UE using the witness (u', v', w'):

$$r'_u \leftarrow r_u + e(u - u')$$
$$r'_v \leftarrow r_v + e(v - v')$$
$$r'_w \leftarrow r_w + e(w - w')$$

Indeed,

$$a' = h_2^{r'_u} \cdot h_3^{r'_v} \cdot h_4^{r'_w} = \frac{h_2^{r_u} \cdot h_3^{r_v} \cdot h_4^{r_w} (h_2^u \cdot \cdot h_3^v \cdot h_4^w)^e}{(h_2^{u'} \cdot h_3^{v'} \cdot h_4^{w'})^e} = a$$

$$z'_u = r'_u + eu' = r_u + e(u - u') + eu' = z_u$$
$$z'_v = r'_v + ev' = r_v + e(v - v') + ev' = z_v$$
$$z'_w = r'_w + ew' = r_w + e(w - w') + ew' = z_w$$

Rephrased: for a different combination of the message (a, z_u, z_v, z_w) sent from an honest UE to a potential cheating HN $\mathcal{A}$, and a possible witness (u', v', w') satisfying $A = h_2^{u'} \cdot h_3^{v'} \cdot h_4^{w'}$, there exist unique values $u', v', w' \in \mathbb{Z}_p$ that satisfy $a = h_2^{r'_u} \cdot h_3^{r'_v} \cdot h_4^{r'_w}$, $z'_u = r'_u + eu', z'_v = r'_v + ev'$ and $z'_w = r'_w + ew'$. This indicates that the above proving process is *witness indistinguishable*.

Zero-knowledge. Now suppose that a dishonest HN is able to determine a witness (u', v', w') after interacting with a UE running the same π_2 polynomially many times. Since this proving process ensures witness indistinguishability, the witness (u', v', w') found by $\mathcal{A}$ will match the witness used by the UE with a probability of exactly $1/p$, where p is a large prime number, e.g., 256-bit. In other words, with a probability close to 1 (i.e., $(1 - 1/p)$), the two witnesses will be different. This can also imply that a cheating prover UE succeeds with probability at most $1/p$, i.e., the prover UE actually knows the witness (u, v, w), which satisfies the *soundness* requirements:

Soundness. If a Probabilistic Polynomial Time (PPT) dishonest UE $\mathcal{U}$ can convince a verifier (i.e., HN) with non-negligible probability using a false statement, then $\mathcal{U}$ can be used to solve the discrete logarithm problem, i.e., it is computationally infeasible for any UE to persuade the HN of the validity of an invalid statement.

Witness Hiding. From another perspective, suppose $\mathcal{A}$ can compute two pairs $(u, v, w) \neq (u', v', w')$ satisfying

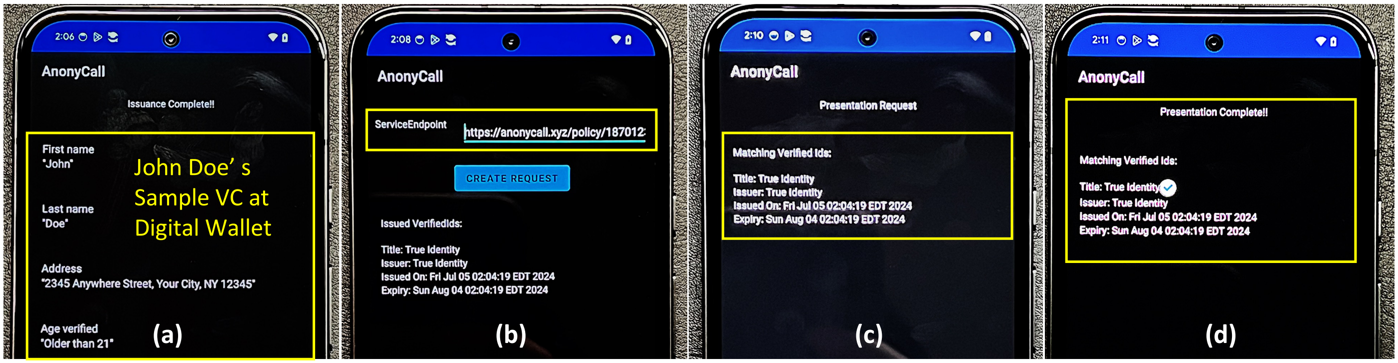
$$A = h_2^u \cdot h_3^v \cdot h_4^w, \quad A = h_2^{u'} \cdot h_3^{v'} \cdot h_4^{w'}$$

which implies that $h_2 = h_3^{\frac{v'-v}{u-u'}} \cdot h_4^{\frac{w'-w}{u-u'}}$. To simplify the notation, we can denote $h_3 = h_4^\lambda$, where the relationship λ is unknown. Transforming the above equation, we get $h_2 = h_4^{\frac{(v'-v)\lambda + (w'-w)}{u-u'}}$. This indicates that $\mathcal{A}$ must solve $log_{h_4} h_2$ to identify the witness, which is as hard as breaking the DL problem. In conclusion, under the DL assumption, this proving protocol is witness hiding, i.e., no PPT HN is able to extract the UE's secret values id, d, r, S_0.

Conclusion. Our proof demonstrates that the charging method preserves the UE's anonymity and unlinkability, provided that no PPT adversary $\mathcal{A}$ can breach it with non-negligible probability. Additionally, the EUF-CMA secure SPS-EQ scheme and the soundness property of the ZKP process ensure that the $cred$ is unforgeable, allowing a UE to pass verification only if it possesses a valid $cred$.

UE		HN
$M_0 = k_0 = 0;\ id, d, r, S_0 \xleftarrow{R} \mathbb{Z}_p;$ $C \leftarrow \mathsf{Commit}(M_0, k_0, id, d, r);$ compute $(C^{S_0}, g_1^{S_0}), \pi_0$.	$\xrightarrow{(C^{S_0}, g_1^{S_0}), \pi_0}$	If π_0 outputs 1, compute $\underline{C_0^{S_0} = C^{S_0} g_1^{S_0 x_0 M} g_1^{S_0 x_1 k}}$; $\sigma \leftarrow \mathsf{Sign}(sk_{HN}, (C_0^{S_0}, g_1^{S_0}))$.
$(C_0, g_1) \leftarrow (C_0^{S_0}, g_1^{S_0})$ $((C_0, g_1), \sigma_0) \leftarrow \mathsf{Adapt}((C_0^{S_0}, g_1^{S_0}), \sigma, \frac{1}{S_0}, pk_{HN}))$ Store $cred_0 : ((C_0, g_1), \sigma_0)$ Prepare for *successor* $cred_1$: $id', d', r', S_1 \xleftarrow{R} \mathbb{Z}_p$; $C_1 \leftarrow \mathsf{Commit}(M, k, id', d', r')$, compute $(C_1^{S_1}, g_1^{S_1})$.	$\xleftarrow{\sigma,\ (C_0^{S_0}, g_1^{S_0})}$	Once per month (payment): $cred_0$ Issuance and Obtain
Compute $c = k \cdot \gamma + d$; $\pi_2 \in ZKP\{(M, k, d, r, id', d', r', S_1) :$ $C_0 = h_0^M \cdot h_1^k \cdot h_2^{id} \cdot h_3^d \cdot h_4^r \quad \wedge$ $C_1^{S_1} = (h_0^M \cdot h_1^k \cdot h_2^{id'} \cdot h_3^{d'} \cdot h_4^{r'})^{S_1} \wedge$ $\underline{M \geq th} \wedge \ \ c = k\gamma + d\}$	$\xleftarrow{\gamma}$ $\xrightarrow{id,\ c,\ cred_0,\ \pi_2}$	$\gamma \xleftarrow{R} \mathbb{Z}_p$ $\mathsf{Verify}(pk_{HN}, (C_0, g_1), \sigma_0)$ Reject if π_2 outputs 0; Otherwise, check for a duplicated id. If a duplicate is found, compute k using Eq. (4) and reject. Grant access iif no duplicate id is found. <u>Subtract</u> m when session ends: $C_1'^{S_1} = C_1^{S_1} \cdot (h_0^m)^{-1}$, $\sigma_1 \leftarrow \mathsf{Sign}(sk_{HN}, (C_1'^{S_1}, g_1^{S_1}))$.
$((C_1', g_1), \sigma_1') \leftarrow \mathsf{Adapt}((C'^{S_1}, g_1^{S_1}), \sigma_1, \mu', pk_{HN}))$ Store $cred_1 : ((C_1', g_1), \sigma_1')$	$\xleftarrow{\sigma_1, (C_1'^{S_1}, g_1^{S_1})}$	Session Establishment Process: Spend $cred_0$ & $cred_1$ Issuance

Fig. 7. Charging Protocol - *Prepaid* Mode (*Postpaid* mode only differs at the three <u>underlined</u> operations.)



Issuance Complete!!

First name
"John"

Last name
"Doe"

Address
"2345 Anywhere Street, Your City, NY 12345"

Age verified
"Older than 21"

ServiceEndpoint https://anonycall.xyz/policy/187012

Issued VerifiedIds:

Title: True Identity
Issuer: True Identity
Issued On: Fri Jul 05 02:04:19 EDT 2024
Expiry: Sun Aug 04 02:04:19 EDT 2024

Presentation Request

Matching Verified Ids:

Title: True Identity
Issuer: True Identity
Issued On: Fri Jul 05 02:04:19 EDT 2024
Expiry: Sun Aug 04 02:04:19 EDT 2024

Presentation Complete!!

Matching Verified Ids:

Title: True Identity ✓
Issuer: True Identity
Issued On: Fri Jul 05 02:04:19 EDT 2024
Expiry: Sun Aug 04 02:04:19 EDT 2024

Fig. 8. Out-of-Band Authentication between Caller and Callee's Agent (VC based Authentication)

2) Proving π_1, π_2: The only difference between π_1(eq. 3) and π_2 (eq. 5) is that π_2 has an additional statement $c = k\gamma{+}d$. We will explain how to prove π_2, which covers π_1. Recall that the objectives for proving π_2 include:
(1) Prove that the M and k in C_0 and $C_1^{S_1}$ are consistent.
(2) Ensure that the k and d used to compute c are consistent with the k and d in C_0.
(3) Verify that the M in C_0 is not less than th.

Objective (1). To prove (1), we can first rewrite C_0 and $C_1^{S_1}$:

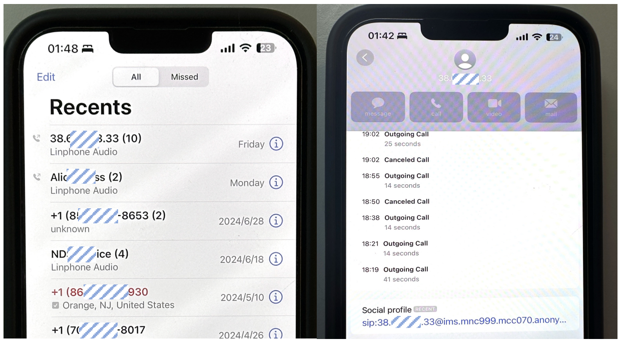
$$C_0 = h_0^M h_1^k h_2^{id} h_3^d h_4^r \quad \rightarrow \quad h_0^M h_1^k = C_0 h_2^{-id} h_3^{-d} h_4^{-r}$$

$$\begin{aligned} C_1^{S_1} &= (C_0 h_2^{-id} h_3^{-d} h_4^{-r} \cdot h_2^{id'} h_3^{d'} h_4^{r'})^{S_1} \\ &= C_0^{S_1} h_2^{S_1(id'-id)} h_3^{S_1(d'-d)} h_4^{S_1(r'-r)} \end{aligned}$$

Recall that C_0 and $C_1^{S_1}$ will be sent to HN during this verification process, so we can treat C_0 as a known base value, similar to h_i. To simplify notation, we denote $A' = C_1^{S_1}$ and represent the randonmizer (i.e., exponents of h_2, h_3, h_4) as $\hat{u}, \hat{v}, \hat{w}$. This meets the objective (1), as this equation implicitly proves that the exponents of h_0, h_1 in $C_1^{S_1}$ are S_0 times those in C_0, with S_0 being a secret value unknown to HN. Due to the hardness of DLP, a UE cannot prove that this equation holds unless it has truly set the exponents to M, k and knows the secret S_0. In essence, the statement we need to prove is

$$A' = C_0^{S_1} \cdot h_2^{\hat{u}} \cdot h_3^{\hat{v}} \cdot h_4^{\hat{w}}$$

The computation and security analysis can be demonstrated using the same method previously described for π_0.



01:48

Edit

All | Missed

Recents

38.[redacted]3.33 (10)
Linphone Audio
Friday

Alic[redacted]ss (2)
Linphone Audio
Monday

+1 (8[redacted]-8653 (2)
unknown
2024/6/28

ND[redacted]ice (4)
Linphone Audio
2024/6/18

+1 (86[redacted]930
Orange, NJ, United States
2024/5/10

+1 (7[redacted]-8017
2024/4/26

01:42

38.[redacted].33

message | call | video | mail

19:02 Outgoing Call
25 seconds

19:02 Canceled Call

18:55 Outgoing Call
14 seconds

18:50 Canceled Call

18:38 Outgoing Call
14 seconds

18:21 Outgoing Call
14 seconds

18:19 Outgoing Call
41 seconds

Social profile RECENT
sip:38.[redacted].33@ims.mnc999.mcc070.anony...

Fig. 9. Call Histories with an Anonymous Callee
* Dial SIP URIs through SIP App (`Linphone`) on iPhone
* Callee's temporary SIP URI: `38.XXX.XX.33@domain.name`

Objective (2). This is equivalent to proving that $c = k\gamma + d$ holds with respect to $C_0 = h_0^M h_1^k h_2^{id} h_3^d h_4^r$. We can eliminate d by using the equation $d = c - k\gamma$, and rewrite $C_0 = h_0^M \cdot h_1^k \cdot h_2^{id} \cdot h_3^{c-k\gamma} \cdot h_4^r$. Recall that M, k, id, d, r are secret values, while the remaining parameters (e.g., c and γ) are known to HN. Thus, we can denote the relation that a UE needs to prove as $C_0 \cdot h_3^{-c} = h_0^M \cdot h_1^k \cdot h_2^{id} \cdot h_3^{-k\gamma} \cdot h_4^r$. We denote $B = C_0 h_3^{-c}$ and $\hat{h_3} = h_3^{-\gamma}$, since C_0, c, γ, h_3 are all known to HN. We can then rewrite the equation to:

$$B = h_0^M \cdot h_1^k \cdot h_2^{id} \cdot (\hat{h_3})^k \cdot h_4^r$$

This objective can now be interpreted as UE needs to prove the knowledge of k, M, id and r and the exponent of base $\hat{h_3}$ is equal to the exponent of h_1 in the above equation. The proving process is similar to π_0 but with the modification that a same randomizer can be used for $h_1, \hat{h_3}$ to prove their equality.

1. Commit. UE generates $r_x, r_u, r_v, r_w \in \mathbb{Z}_p$ and computes

$$a = h_0^{r_x} \cdot h_1^{r_v} \cdot h_2^{r_u} \cdot \hat{h_3}^{r_v} \cdot h_4^{r_w}$$

2. Challenge and response. UE computes $e = H(B||a)$ and

$$z_x = r_x + eM, \quad z_u = r_u + e \cdot id, \quad z_v = r_v + ek, \quad z_w = r_w + er$$

UE sends (a, z_x, z_u, z_v, z_w) to the HN.

3. Verification. HN computes $B = C_0 \cdot h_3^{-c}$, $\hat{h_3} = h_3^{-\gamma}$, and a challenge $e' = H(B||a)$ independently, and checks if the equation holds: $B^{e'} \cdot a \overset{?}{=} h_0^{z_x} \cdot h_1^{z_v} \cdot h_2^{z_u} \cdot h_3^{z_v} \cdot h_4^{z_w}$. This process convinces the HN that the UE successfully computed $c = k\gamma + d$ based on C_0 without revealing k, M, id, or r.

Objective (3). Range proof is a form of ZKP that allows a prover to convince a verifier that a committed value lies within a range without revealing the value itself. We now explain how a UE can use range proof to prove that the M committed in C_0 is not less than th. We set $th = 5$ minutes as an example, but it may vary due to different policies defined by each operator. Since range proofs handle ranges $[0, 2^n - 1]$, to prove $M \geq 5$, we can equivalently show that $M' = M - 5$ is in the range $[0, 2^n - 1]$ for some suitably large n (e.g., $n = 10$ is big enough to cover around 1000 minutes per month). We then rewrite the commitment as $C_0 h_0^{-5} = h_0^{M'} h_1^k h_2^{id} h_3^d h_4^r$. Thus, our objective is to prove that given this adjusted commitment, the exponent of h_0 is non-negative. We can eliminate the influence of other secrets k, id, d, r and improve the computation efficiency by generating a dedicated commitment for M' as $C'_M = h_0^{M'} h_4^r$, and prove the M' is non-negative, then add a simple proof to show that the M' in $C_{M'}$ is equal to the M in $C_0 h_0^{-5}$, which can be proved using the same technique applied in objectives (1)(2). The process of this range proof:

1. Bit Decomposition. First, we denote M' as a binary representation $M' = \sum_{i=0}^{n-1} x_i 2^i$, with $x_i \in \{0, 1\}$. In fact, our previous objectives involved proving the case for $n = 1$, now we decompose the proving of M' in a bit-by-bit manner.

2. Commit to each bit. Then compute a sets of Pedersen commitments to commit to each bit of M': $B_i = g^{x_i} h^{r_i}$, where $r^i \in \mathbb{Z}_p$ is a random blinding factor for each bit. As long as we can prove each x_i is either 0 or 1 using the NIZK schnnor proof techniques that we introduced previously, we can prove that the M' is non-negative.

3. Schnorr proofs for each bit. Finally, a UE can show to the HN that the commitments to the binary bits reconstruct the original commitment, which can be done by combining the commitments $C'_M = \prod_{i=0}^{n-1} (B_i)^{2^i}$. HN can verify if the commitment is correct by computing the addition independently.

In essence, this range proof involves running the previous NIZK proof in parallel according to the number of bits n. Thus, we omit the detailed proving process. The efficiency can be optimized by using techniques such as Bulletproofs [20].

C. Discussion: Answer for Q5

What if a malicious HN intercepts the call session? (Potentials to support E2EE) In many jurisdictions, MNOs do not directly intercept communication content, as lawful interception is tightly regulated and permitted only under specific legal authorization. However, if an MNO behaves maliciously, AnonyCall can be extended to support E2EE communication to protect against malicious interception via our out-of-band authentication method. Here we provide a succinct vision. E2EE relies on exchanging public keys between the caller and callee to derive a shared session key, ensuring that only the intended endpoints can decrypt the communication and protecting content from untrusted service providers. While E2EE is not natively supported in cellular networks due to the complexity of managing keys across billions of devices and multiple carriers, AnonyCall enables a practical alternative. Specifically, callers and callees can leverage the existing trust frameworks discussed in Sec. II-C to negotiate a shared session key out-of-band and perform encryption and decryption directly at their IMS clients. A SIP header parameter can indicate the encryption scheme in use, allowing the callee's client to process encrypted media accordingly.

We note that there are numerous third-party VoIP applications (e.g., Signal, WhatsApp, Telegram, Skype, Zoom) that offer E2EE mechanisms to safeguard multimedia content. However, these platforms require all users to operate on the same or compatible systems that support the same security protocols. They also do not protect caller/callee identity privacy since they are fundamentally incompatible with anonymous access methods due to similar challenges in anonymous callee discovery—when users remain anonymous, service providers cannot relay the correct public keys necessary for E2EE.

Appendix B
Artifact Appendix

This appendix accompanies the paper and describes the publicly released artifact for AnonyCall. The artifact demonstrates an IMS-compatible SIP architecture with out-of-band caller authentication and a privacy-preserving charging protocol. All source code, scripts, and documentation are publicly available via Zenodo at: https://zenodo.org/records/17851159.

This appendix targets general readers. It provides a concise overview of the artifact and its reproducibility, while detailed setup instructions are deferred to the accompanying README.

A. Description & Requirements

The artifact supports reproducing the paper's core system functionality, including SIP call establishment with temporary identifiers, out-of-band authentication, and usage-based charging. The README was originally prepared for the Artifact Evaluation (AE) process and references an evaluation-specific deployment. For public use, readers are encouraged to follow the updated instructions to deploy a standard IMS core locally.

1) Access Model:

- **Artifact bundle.** The Zenodo archive contains this appendix, the updated README, screenshots, and the complete `Anonycall_code/` tree.
- **IMS core.** A Google Cloud Platform (GCP) VM was used during the AE process but may no longer be available. Reproducibility does not depend on this VM. Readers should deploy a standard IMS server locally using open-source components (e.g., Kamailio), as described in the README.
- **Code layout.** The artifact includes components for authentication services, Android clients, charging logic, and orchestration scripts. Each directory contains its own documentation.

2) Hardware Dependencies:

- A commodity workstation with 8–16 GB RAM and sufficient disk space for Android tooling.
- One or two SIP-capable endpoints (e.g., smartphones).

3) Software Dependencies:

- Android Studio with emulator support;
- OpenJDK 17;
- Python 3.9 or newer;
- OpenSSL;
- SIP clients such as Zoiper or Linphone.

Exact version requirements and installation commands are provided in the README.

4) Benchmarks: No third-party datasets are required. The paper's quantitative results are derived from SIP signaling traces and charging benchmarks generated by the artifact.

B. Artifact Installation & Configuration

Readers should follow the README to (i) deploy a IMS core, (ii) configure SIP clients, and (iii) build and run the Android-based authentication and charging components. These steps mirror the setup used in the paper, except that the IMS core runs locally rather than on an evaluation-specific cloud VM.

C. Experiment Workflow

- **Part I – SIP call reproduction.** Establish SIP sessions through the local IMS core and inspect signaling traces.
- **Part II – Out-of-band authentication and charging.** Execute the provided workflows to demonstrate caller authentication and usage-based charging.

D. Major Claims

- (C1) **Temporary SIP URIs interoperate with a standard IMS core without SIP protocol changes.**
- (C2) **Out-of-band caller authentication supports both PKI and DID/VC policies with sub-second latency.**
- (C3) **The privacy-preserving charging protocol enforces usage-based billing with millisecond-level cryptographic overhead.**

E. Evaluation

All experiments follow the same structure as used during the AE process, with the only difference being that the IMS core is deployed locally.

1) Experiment (E1) – SIP Call Reproduction: Goal: Validate that temporary SIP identifiers function correctly on a standard IMS core.

Execution. Register two SIP clients with the locally deployed IMS server and establish a call session. Inspect SIP traces to confirm correct call setup and teardown.

2) Experiment (E2) – Out-of-Band Authentication: Goal: Demonstrate PKI-based and DID/VC-based authentication workflows.

Execution. Run the Android demos using the local authentication services. Successful runs retrieve the callee's SIP URI and enable call establishment.

3) Experiment (E3) – Charging Protocol: Goal: Reproduce the cryptographic performance results reported in the paper.

Execution. Execute the charging workflows and record timing results for credential issuance and usage.

4) Experiment (E4) – Distributed Authentication (Optional): Goal: Evaluate authentication latency under realistic network conditions.

Execution. Optionally deploy the authentication service on a remote host and repeat Experiment E2.

F. Notes

Performance results may vary due to hardware and network conditions. All core functionality can be reproduced without access to private infrastructure or paid cloud services. The Zenodo archive serves as the long-term reference for the artifact.