NASCENT: Tackling Caller-ID Spoofing in 4G Networks via Efficient Network-Assisted Validation

Amit Sheoran, Sonia Fahmy, Chunyi Peng, Navin Modi
Department of Computer Science, Purdue University, West Lafayette, IN, USA
E-mail: {asheoran, fahmy, chunyi, modin}@purdue.edu

***Abstract*— Caller-ID spoofing deceives the callee into believing a call is originating from another user. Spoofing has been strategically used in the now-pervasive telephone fraud, causing substantial monetary loss and sensitive data leakage. Unfortunately, caller-ID spoofing is feasible even when user authentication is in place. State-of-the-art solutions either exhibit high overhead or require extensive upgrades, and thus are unlikely to be deployed in the near future. In this paper, we seek an effective and efficient solution for 4G (and conceptually 5G) carrier networks to detect (and block) caller-ID spoofing. Specifically, we propose NASCENT, Network-assisted caller ID authentication, to validate the caller-ID used during call setup which may not match the previously-authenticated ID. NASCENT functionality is split between data-plane gateways and call control session functions. By leveraging existing communication interfaces between the two and authentication data already available at the gateways, NASCENT only requires small, standard-compatible patches to the existing 4G infrastructure. We prototype and experimentally evaluate three variants of NASCENT in traditional and Network Functions Virtualization (NFV) deployments. We demonstrate that NASCENT significantly reduces overhead compared to the state-of-the-art, without sacrificing effectiveness.**

I. INTRODUCTION

Vulnerabilities in widely-deployed packet-based telecommunications services have raised serious concerns about the security of current infrastructure [1]. A simple (and now pervasive) type of attack that exploits 4G Voice over LTE (VoLTE) vulnerabilities is the caller-ID spoofing attack [2], where an attacker impersonates another user by spoofing their telephone number or user name. An unsuspecting user may be deceived by the spoofed caller-ID displayed by their user equipment (UE) since this ID can correspond to a trusted organization such as a government agency [3]. Telemarketers also often use caller-ID spoofing to avoid detection by caller identification systems (e.g., Truecaller [4]), and trick users into receiving marketing calls. A recent phenomenon, neighbor spoofing [5], [2], uses a caller-ID that closely matches the receiver telephone number.

While caller-ID spoofing attacks were difficult to mount on traditional circuit-switched networks, the proliferation of SIP-based VoLTE services and easy access to caller-ID spoofing applications (e.g., SpoofCard [6] and SpoofTel [7]) have enabled an average telephony subscriber to mount such attacks, leading to losses in the billions of dollars [5].

Fundamentally, the caller-ID spoofing attack stems from a well-known vulnerability in the IP Multimedia Subsystem (IMS). Traditional IMS servers designed for Voice over IP (VoIP) do not validate the subscriber identifier in incoming call setup requests, which allows an attacker to impersonate other subscribers. Even if IMS servers can validate the caller-ID of incoming calls, the IMS network alone does not have sufficient information to validate the caller-ID [8]. In the case of VoLTE, a user is initially authenticated, but the identity indicated in the call setup requests arriving later is not validated by the IMS.

Several solutions have been proposed to tackle caller-ID spoofing. These include network-assisted authentication using shared secrets and cryptographic encryption [9], end-to-end certificate authentication [10], [11], [12], [13], challenge-response authentication (between caller and callee) [14], and call-back validation [15], [16]. Unfortunately, these solutions suffer from several drawbacks. Encryption-based solutions require additional message exchange with endpoints and expensive encryption. Certificate-based authentication requires additional infrastructure to manage and validate certificates. Call-back systems generate a validation call towards the caller-ID of an incoming call, effectively doubling the signaling workload. All endpoint-only approaches suffer from the problems that endpoints cannot always be trusted, and that a massive number of endpoints would need upgrade. These drawbacks ultimately make current solutions ineffective or infeasible to deploy. This leads us to focus our attention on designing *network-assisted solutions that are efficient and easy-to-deploy*.

We design a network-assisted approach to detect caller-ID spoofing, NASCENT (Network-assisted caller ID authentication). By sharing intelligence between the Evolved Packet Core (EPC) and IMS networks, carriers can efficiently and effectively detect caller-ID spoofing at runtime, without requiring major infrastructure deployment or endpoint upgrades. We leverage subscriber data already available to EPC control-plane functions, but cross validate the caller-ID of an incoming voice call *at the IMS* to reduce the overhead on the EPC data-plane. We make the following contributions:

1) We propose NASCENT, a new lightweight spoofing detection approach that is easy-to-deploy in 4G and beyond.
2) We develop prototypes of three variants of NASCENT.
3) We experimentally evaluate the performance of NASCENT variants, and compare them to the RFC-defined proxy-to-user authentication [9] in both traditional and Network Functions Virtualization (NFV) de-

ployments. We demonstrate that NASCENT is effective and exhibits low overhead.

The remainder of this paper is organized as follows. In §II, we describe VoLTE, caller-ID spoofing, and related work. In §III, we compare prior network-assisted approaches to counter caller-ID spoofing. In §IV, we discuss the design of our new approach, NASCENT, and in §V, we experimentally evaluate NASCENT. In §VI, we discuss deployment of NASCENT, and §VII concludes the paper.

II. BACKGROUND AND RELATED WORK

4G LTE (and beyond) advance cellular networks to a packet-switched only infrastructure, migrating traditional circuited-switched voice support to VoLTE [17]. VoLTE carries voice traffic and its signaling in IP packets, akin to VoIP. In this section, we introduce necessary VoLTE background and explain why caller-ID spoofing is possible even with authentication in cellular networks. Finally, we summarize related work on countering caller-ID spoofing.

VoLTE architecture and call setup. Figure 1 depicts a simplified LTE network architecture and the VoLTE call setup flow. LTE provides voice service to user equipment (UEs, i.e., phones) in its core network, which consists of two main subsystems: Evolved Packet Core (EPC) and IP Multimedia Subsystem (IMS). EPC is responsible for data-plane packet delivery and its associated control functions such as the Policy and Charging Rules Function (PCRF), user authentication, and security. The Packet Data Network Gateway (PGW) is the EPC's critical network function which forwards packets and acts as the interface to other packet data networks like the Internet and IMS. The PGW typically includes the control function commonly known as the Policy and Charging Enforcement Function (PCEF), which communicates with the PCRF for quality and billing policy enforcement. The IMS offers voice and multimedia services over IP via Call Session Control Functions (CSCFs). IMS uses the Session Initiation Protocol (SIP) [9] for call setup signaling, which is the standard for VoIP.

A caller's UE must authenticate itself before making a call (step 1). User authentication is performed when the UE initially attaches to the network (e.g., powers on). Each UE's SIM card is associated with an International Mobile Subscriber Identity (IMSI) and a Mobile Station International Subscriber Directory Number (MSISDN) (telephone number), which are globally unique. A UE secret key is stored at the Home Subscriber Server (HSS), a user database. The Mobility Management Entity (MME) enforces user authentication towards the HSS, and updates authenticated UE information at the PGW. After that, the UE is authorized to make a call (step 2). To initiate a call, the UE sends a call setup request in a `SIP INVITE` message to the IMS which forwards the request to the callee. IMS later performs authentication and authorization (AA) with the PCRF (2d) and finally with the PGW (2e) using the Diameter protocol [18]. This is needed for charging and QoS policy control. We show the signaling flow as a space-time diagram in Figure 3a.

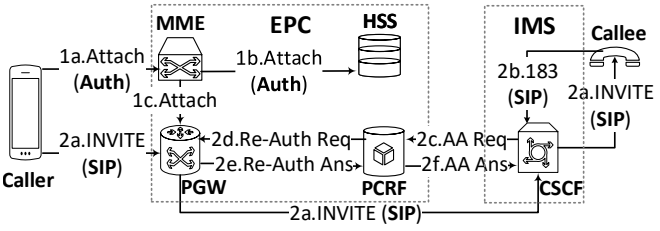


Fig. 1: LTE network architecture and VoLTE call setup flow.

Caller-ID spoofing. Caller-ID spoofing is feasible in VoLTE despite user authentication [1], [19], [20], [21]. The IMS and EPC use different addressing mechanisms to identify a UE. In IMS, the caller-ID is carried in the `From` header in the `INVITE` message. This header denotes the authentic caller's telephone number in the case of no spoofing. However, there is no guarantee that the forwarded caller-ID in the (`INVITE`) is exactly the same as the one which was authenticated in advance (IMSI and its true phone number) or associated with the derived one (e.g., temporary ID or the IP address allocated). In fact, real-world experiments have already validated that the current practice does not enforce any binding between SIP IDs and authenticated IDs, making users vulnerable to caller-ID spoofing [1], [19], [20], [21]. The root cause of caller-ID spoofing lies in the separation between user authentication and call setup signaling. Although authentication is initially executed (to authorize making a call), no mechanism prevents the caller from later altering the forwarded ID, thus hiding its authenticated ID during call setup.

Related work. Several solutions have been recently proposed in the literature. These can be categorized as endpoint-only or network-assisted. Some endpoint-only solutions [15], [14] use challenge-and-response between the caller and callee, which requires the caller to respond to an SMS [14] or a call [15]. This requires the caller's cooperation, and mandates updates on all phones (i.e., all possible callers), which is unlikely in the foreseen future. Most network-assisted solutions either deploy an additional global authority (e.g., a public certification service [10], [22], [23], [24]) or a Public Key Infrastructure (PKI) [25] to authenticate each party before call setup. An easier-to-deploy approach is to authenticate callers at the gateway during call setup [8], [26] by cross validating the forwarded ID with the authenticated one. This approach is effective in principle but has not been deployed in practice, partly because all existing solutions would incur an unacceptable performance penalty. Our work adopts this general approach but designs a practical solution compatible with current infrastructure at a much lower overhead.

III. DESIGN GOALS AND LESSONS LEARNED

In this work, we aim to develop practical spoofing detection in carrier networks. We believe that detecting caller-ID spoofing *with network-assistance* is more effective and easier-to-deploy. This is because carrier networks are under the control of a few trustworthy service providers, which wish to protect users from ill-intended spoofing abuse, and

TABLE I: Comparison of network-assisted caller-ID spoofing detection solutions.

Solution	Effectiveness		Ease of deployment		Overhead			
	SIP Spoofed	SIP & IP Spoofed	Infra-structure	Standards-Compatibility	Network # Core	Network # UE	Computation	Storage
[RFC] Proxy-to-user authentication [9]	✔	✔	None	Yes	6	6	High	Low
[RFC] TLS [9]	✔	✔	PKI	Yes	5	5	High	High
Passive validation [21]	✔	✘	Not applicable					
iVisher [8] •	✔	✔	None	No	21	0	Low	Low
Kim et al. [26]	✔	✔	None	No	0/(8 ◇)	0	High	High
NASCENT	✔	✔	None	Yes∗	0/4/6∗	0	Low	Low

• Works for VoLTE and VoIP; ◇ When stored in a remote key-value store; ∗ Depends on variant used

enforce authentication and authorization, as commonly expected. In this section, we present our design goals and compare existing *network-assisted* approaches. Our objective is to understand the pros and cons of current solutions and gain insights for the design of NASCENT in §IV.

A. Goals

An ideal network-assisted solution should be *effective*, *easy-to-deploy* and *efficient-to-run*.

(1) Effectiveness. An effective solution should detect both simplistic and sophisticated attacks. In the simplest case, the caller-ID in the `INVITE` From header is forged. An effective solution must work when the attacker spoofs other caller-IDs carried in the `From`, `To`, or `P-Asserted-Identity` fields, as well as the IP address. Note that when SIP messages are tunneled using other protocols, the source/destination IP address can be easily spoofed without impacting end-to-end packet delivery.

(2) Ease of deployment. An easy-to-deploy solution requires minimal hardware and software upgrades to the existing infrastructure. Solutions should not require (i) additional infrastructure such as PKI, or (ii) non-standard protocols or interfaces. A desirable solution should leverage existing, standard-compatible components and only require software upgrades.

(3) Efficiency. An efficient solution should exhibit low overhead in three aspects. **(i) Network overhead** refers to additional message exchanges required to support caller-ID spoofing detection. This includes: (a) Messages exchanged between network functions (NFs) within IMS or EPC, and (b) Messages exchanged between the UE and the EPC and IMS NFs. Since the EPC and IMS networks are often co-located or connected via high-speed links, message exchange between these NFs traverses fewer hops than message exchange between the UE and core network (IMS and EPC). Traversal of more hops, coupled with the latency introduced by last-mile radio links, makes message exchange with a UE more expensive. In the core, we count the logical number of messages exchanged between NFs. In practice, NFs may be connected via multiple hops, or the functionality of an NF may be collectively implemented by multiple nodes. **(ii) Computation overhead** refers to overhead of message processing, e.g., cryptographic calculations have higher overhead compared to trivial comparisons. **(iii) Storage overhead** refers to memory and disk usage. Since the precise computation and storage overhead depends on the implementation and deployment model, we only classify these overheads as high or low in Table I, but they highly affect our results for both the PGW and IMS in §V.

B. Comparison of Existing Proposals and Lessons Learned

We compare existing network-assisted solutions in Table I.

The standard (RFC 3261) [9] proposes two runtime caller-ID validation approaches: a challenge-response procedure (proxy-to-user authentication) and an encrypted channel in Transport Layer Security (TLS). Both are deemed effective but not efficient or easy-to-deploy because they require additional infrastructure, exchange additional messages with the endpoints, or involve expensive computations for decryption.

Passive validation [21] checks the caller-ID in the `INVITE` request only and thus is ineffective when the attacker spoofs both the IP address and the SIP header. For this reason, we do not consider it further. Some proposals utilize control-plane information available at network gateways to validate the caller-ID. iVisher [8] validates the caller-ID by tracing the call back to the originating gateway. While effective, iVisher requires several new messages which are not standard-compatible and thus require substantial upgrades at the gateways. An alternative solution [26] detects caller-ID spoofing by inspecting every SIP message received at the EPC gateway (e.g., PGW). This incurs high computation and storage overhead due to deep packet inspection, as the PGW is responsible for forwarding all IP packets, not just `SIP INVITE`. It is also expensive for the PGW to encode SIP protocol messages and terminate data-plane connections – operations typically performed by the CSCF – since the PGW is not SIP-aware.

Lessons learned. The above discussion sheds light on designing an effective, standard-compatible, low-overhead solution. First, the solution should leverage existing infrastructure and should purely be a software solution. Second, limiting the entire solution to a single data-path network function induces unacceptable overhead. The EPC gateway has the user authentication information needed for network-assisted validation but it lacks the context of VoLTE call setup. A gateway-only solution has a high computation cost (deep packet inspection) and resource waste (most packets are not VoLTE relevant). An IMS-only solution is infeasible

since the IMS does not have authentication data to validate a caller-ID. Third, overhead of network communication with the endpoints is much higher communication within the core network, since messages to endpoints traverse lossy last-mile radio links and experience higher latency and more failures. Fourth, communication between network functions should exploit existing protocols and interfaces; otherwise, it is not standard-compatible and is more difficult to deploy (patch existing infrastructure).

IV. NASCENT DESIGN

Based on the goals in §III-A, we need to design an effective, low-overhead and easy-to-deploy caller-ID spoofing detection solution that does not suffer from the drawbacks of the state-of-the-art network-assisted approaches discussed in §III-B.

A. NASCENT *Overview*

Our solution, NASCENT, uses a *cross validation* approach. Unlike passive identifier validation solutions [21] that only utilize information available to the IMS servers, *cross validation* compares UE identifiers from *multiple networks*: the EPC and IMS networks in our case. The idea of cross validation stems from the availability of at least one authenticated network identifier that can be reliably used to identify a network endpoint.

We make the following key decision in designing NASCENT: *We split the caller-ID cross validation functionality among the IMS control plane and the PGW. We minimize expensive operations at the PGW, in order to reduce latency and overhead.* Since IMS servers already manage and terminate SIP sessions, they require minimal changes to implement caller-ID validation. As shown in Figure 3a, the EPC network already supports communication between the IMS servers and EPC packet gateways [11], [27], [28]. Figure 2 depicts the basic idea of NASCENT. The PGW creates a mapping of the EPC identifiers (e.g., MSISDN) and IMS identifiers (e.g., SIP `Call-ID` [9], `From`) when it receives an `INVITE` message (step 1a). Before forwarding the `INVITE` request to the called UE, the IMS fetches the EPC identifier associated with the `INVITE` message (step 3 and 3a) and cross validates the caller-ID being forwarded against the MSISDN received from the EPC. Figure 2 depicts a simplified view of a traditional deployment. In practice, however, the EPC and IMS functions can be decomposed and deployed as multiple Virtualized Network Functions (VNFs) or can be aggregated and deployed as a single VNF, which does not impact our design.

NASCENT consists of following three components (new steps highlighted in blue in Figure 2):

(1) Mapping creation. The PGW monitors SIP messages generated by a UE and stores a mapping between the IMS and EPC identifiers when a SIP `INVITE` message is observed. The PGW already extracts the SIP payload from each tunneled packet and forwards this payload to the IMS servers. The PGW typically allocates a dedicated network interface (Access Point Name (APN)) for IMS signaling messages

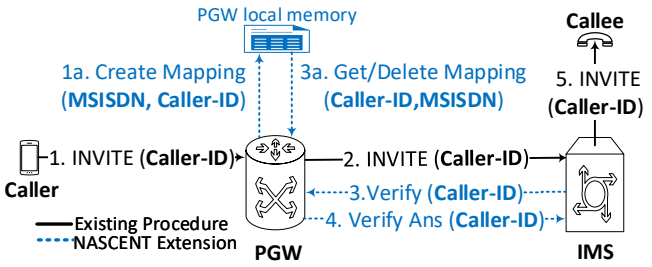


Fig. 2: NASCENT design: 1a and 3a are used to access local memory (i.e., no messages are exchanged).

and therefore SIP traffic can be efficiently monitored by observing traffic on this interface. The `Call-ID` header can be used by the PGW and IMS to uniquely identify a SIP message. (The actual headers/parameters used by the PGW and IMS to identify a SIP message depend on the implementation.)

The PGW will extract the SIP headers (`Call-ID`, `From`, `To`) and IP address, and save a mapping between these headers and the EPC identifiers (MSISDN, IMSI) associated with the tunnel. This is effective because the EPC network uses data tunnels to transport VoLTE signaling messages between the IMS and UE. We utilize the knowledge of tunnel identifiers associated with a UE to validate the UE identity in SIP signaling messages. The tunnel identifiers in EPC are used to transfer encrypted traffic between the PGW and UEs, and are unchanged for the duration a user session. This property of tunnel identifiers allows us to reliably associate each SIP request with a trusted identifier (MSISDN), using which runtime validation of caller-ID can be performed.

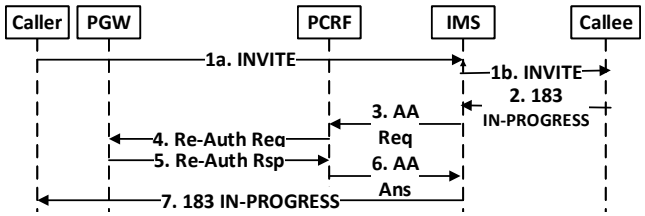
(2) Caller-ID validation. The IMS server CSCF queries the PGW for the EPC identifiers associated with a SIP `INVITE` message and validates the SIP headers (e.g., `From`, `To`) against the EPC identifiers. Since the PGW is configured to store the mapping of SIP headers and EPC identifiers, the CSCF uses the value extracted from the `INVITE` message to generate a validation request towards the PGW. The EPC network already provides well-defined, standard-compatible interfaces to communicate with IMS, and hence these interfaces can be leveraged for this operation.

(3) Mapping deletion. After replying to the CSCF, the PGW deletes the EPC and IMS identifier map for this caller-ID. Implicit deletion reduces memory requirements at the PGW since each mapping is only stored for a few milliseconds.

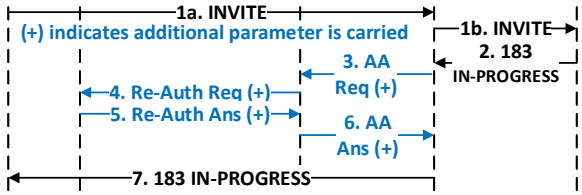
B. NASCENT *Variants*

The current VoLTE architecture presents two main challenges to the design of NASCENT:

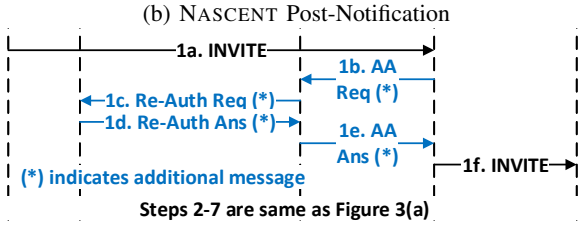
(1) The IMS AA procedure is performed after the callee is notified. As shown in Figure 3a, the IMS server only triggers rule generation after receiving media information from both caller and callee (from step 1a and step 2). Without additional signaling messages, the network can only detect a spoofed call after the user is notified of an incoming voice call (post-notification). Even if a spoofed call is detected



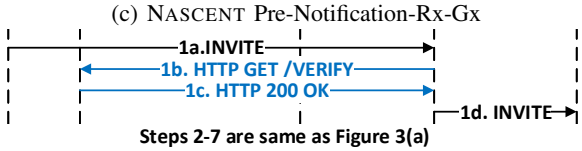
(a) Existing control plane interactions for a VoLTE call



(b) NASCENT Post-Notification



(c) NASCENT Pre-Notification-Rx-Gx



(d) NASCENT Pre-Notification-Rx+

Fig. 3: NASCENT variants.

and terminated by the network, the network has no means of conveying this information to the callee, and the user would still receive a "missed call." The network can convey the spoofed call notification to the user via a SIP CANCEL message used to terminate a spoofed call, and the UEs can be upgraded to support this spoofed call notification mechanism. Optionally, the operators can employ an external notification mechanism (such as SMS) to convey the spoofed call alert. If the percentage of spoofed calls in the network is relatively low, this may be acceptable.

(2) There is no direct communication between the IMS and PGW. If the network can validate the caller identity before the voice call is forwarded to the callee (pre-notification), spurious notifications can be avoided. In this case, the IMS network must query the PGW. IMS-to-PGW communication is mediated by the PCRF (Figure 3a). The IMS network uses the Diameter Rx interface [28] to exchange messages with the PCRF. The PCRF forwards messages to the PGW using the Diameter Gx [27] interface. A more efficient way to exchange EPC identifier information is to allow the IMS network to directly query the PGW by adding a new interface.

We therefore explore three alternative designs based on (a) whether the caller is validated before forwarding the voice call to the callee, and (b) if the EPC identifier information is queried using the existing Rx-Gx interface, or a new interface is added between the IMS and the PGW. These NASCENT variants are summarized as follows.

(1) Post-Notification. No explicit messages are exchanged between the PGW and IMS to detect a spoofed call. The PGW provides the EPC identifiers to the IMS during the normal procedure after the user receives the voice call (Figure 3b). Rx and Gx messages can be modified to tunnel the additional parameters required to detect spoofing. The callee may receive a missed call notification when this variant is deployed.

(2) Pre-Notification-Rx-Gx. Caller-ID validation uses new signaling messages exchanged between the PGW and IMS prior to the `INVITE` message being forwarded to the callee. The PGW and IMS communicate using existing Rx and Gx interface messages and no new interfaces are required. Additional messages (Figure 3c) relayed via the PCRF incur networking overhead but avoid maintaining additional configurations and connections at the PGW and IMS.

(3) Pre-Notification-Rx+. Caller-ID validation uses a new REST interface between the IMS and PGW. As shown in Figure 3d, the IMS uses this new interface to validate the caller identity before forwarding the message to the callee. This incurs configuration overhead as it requires the IMS to directly communicate with the PGW that is currently serving a user, and the IMS must therefore maintain a list of currently active PGW instances in the network.

C. Meeting Design Goals

NASCENT meets the goals of effectiveness, ease-of-deployment, and low overhead discussed in §III-A as follows (see last row in Table I): (a) NASCENT is effective with sophisticated spoofing attacks through its use of tunnel identifiers, (b) NASCENT does not use PKI, does not define new protocol messages and is compatible with the standards, (c) All NASCENT variants only require few additional messages, all between NFs in the core, thus exhibiting low network overhead, (d) NASCENT does not communicate with endpoints, reducing latency and overhead, (e) NASCENT only requires the PGW to provide the EPC identifiers associated with an `INVITE` message, and does not require the PGW to handle SIP request/response messages or terminate transport-layer connections initiated by the UE, thus incurring low computation overhead, and (f) NASCENT only requires the PGW to maintain each EPC and IMS identifier mapping for a brief period of time (until the call is accepted/rejected) and therefore does not require significant storage at the PGW.

V. EXPERIMENTAL EVALUATION

In this section, we quantify the throughput, resource utilization, and latency incurred in VoLTE call setup.

A. Implementation and Experimental Setup

We have developed a prototype of the IMS CSCF, PCRF and PGW to emulate VoLTE calls in our test environment as shown in Figure 4. The IMS consists of a SIP server that is used for handling SIP messages from the endpoints, and a policy module. We use Kamailio [29] version 5.0.4

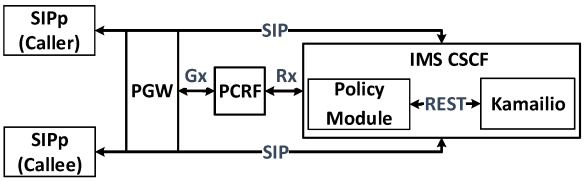


Fig. 4: Experimental setup.

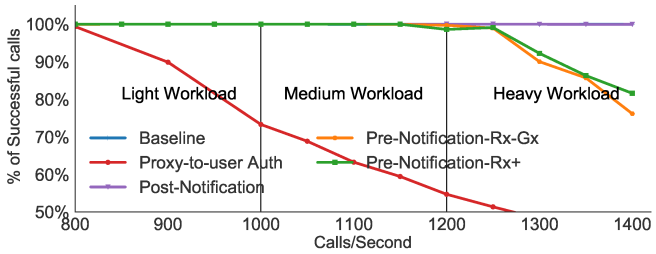


Fig. 5: Percentage of successful calls with 0% spoofed calls.

as the SIP server. We extend the functionality of Kamailio to use a REST message interface to communicate with the policy module. The policy module supports REST interfaces using which Kamailio can trigger Diameter Rx Interface functions [28], [30] to communicate with the PCRF. The PCRF communicates with the PGW using the Diameter Gx [27] Interface. The policy module supports the REST interface using using the KORE library [31] (version 2.0.0). The policy module, PCRF, and PGW are developed as application extensions in the FreeDiameter library [32] version 1.2.1 using the C language (∼3700 lines of new code).

We compare proxy-to-user authentication (§III-B) and the proposed NASCENT variants with a baseline in which the caller-ID is not validated. We select proxy-to-user authentication as a representative network-assisted solution because (a) This approach is specified by the SIP RFC [33] and is already supported by existing implementations, and (b) Previous work [34] has found that its throughput is higher than TLS-based solutions.

In the baseline case, the PGW does not intercept SIP traffic from the UE, and Rx and Gx interface messages do not carry additional EPC identifiers. Proxy-to-user authentication is similar to the baseline case but uses additional messages to authenticate callers using the procedure defined in [9].

We use Docker version 17.03.0-ce and Docker-compose (v1.11.2) [35] to deploy and manage Virtualized Network Functions (VNFs) as shown in Figure 4. Each VNF runs within a container, and all containers are deployed on the same physical host: a Dell PowerEdge R430 (2x Intel Xeon E5-2620 v4) with 16 cores and 64 GB RAM.

Deployment models. We evaluate two deployment models: (a) Traditional deployment, and (b) Network Functions Virtualization (NFV) deployment. In traditional deployment, IMS and EPC are independent physical systems and no resources are shared among them. This setup emulates current deployments where IMS are EPC are deployed on separate physical machines. Kamailio is allocated a single core on CPU-1, while the policy module, PCRF, and PGW share the second CPU. This setup is used to measure the additional resources required to support the caller-ID spoofing solutions on the IMS servers. In NFV deployment, we instantiate all VNFs in Figure 4 on the same physical machine and configure them to share 4 cores on CPU-1. This is akin to expected 5G deployments.

Workload generation. We deploy two instances of SIPp [33], each on a separate physical machine. One SIPp instance is used as the caller and the other is used as the callee. Both caller and callee SIPp instances register the UEs with the IMS prior to the generation of `INVITE` messages. We observe the response codes received by SIPp and use them to infer the number of failures. Per the SIP specification, only `200 OK` messages indicate success and all other response codes are considered failures. The timeout for `INVITE` messages is set to 1 second; that is, an `INVITE` call is considered successful if a `200 OK` response is received within 1 second.

To generate workloads where the caller-ID is spoofed by the SIP caller, we configure SIPp to use a random value in the `From` header of the `INVITE` message. Rejection of a voice call with a spoofed caller-ID is considered a successful result. Therefore, in the plots in §V-B, we count `INVITE` calls rejected due to caller-ID mismatch as successful calls.

Our experiments aim at quantifying the overhead of caller-ID validation on the performance of VoLTE. In real deployment scenarios, EPC networks are over-provisioned to handle flash workloads, and therefore, rarely, if ever, reach actual capacity. We thus use simulated workloads to study NASCENT under a wide range of loads from light to heavy (for stress testing). In both deployment models, we increase the workload until we saturate available system resources. We generate workloads between 800 calls/s to 2000 calls/s. We record the number of successful calls, CPU utilization, and time taken by the IMS to successfully process a voice call request, i.e., time taken to send a `200 OK` response code after an `INVITE` request is sent. We also measure the impact of the percentage of spoofed calls on performance. We generate workloads where 0-10% of `INVITE` message have a spoofed caller-ID. Each experiment runs for 30 seconds and the results represent the mean of at least 10 samples for each experiment. We also compute the standard deviation among the values. The standard deviation was within 1% of successful call percentage in the figures in §V-B. We will note the standard deviation for call setup latency where relevant. We use `docker stats` [35] to measure the CPU usage of VNFs. CPU usage is monitored every second and our plots represent the average CPU utilization over the experiment duration.

B. Experimental Results

1) Traditional Deployment Model: We begin by benchmarking the performance of the VoLTE calls in the baseline setup when no caller-IDs are spoofed. We compare the number of successful calls for each caller-ID validation solution to the baseline results.

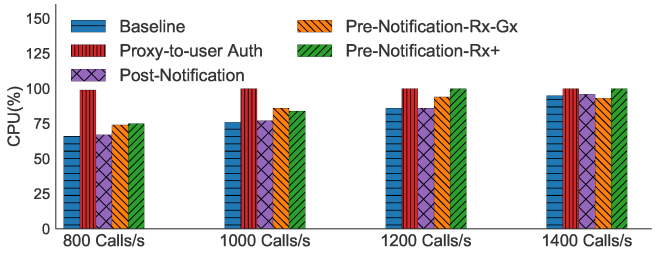


Fig. 6: CPU utilization of IMS Server with 0% spoofed calls.

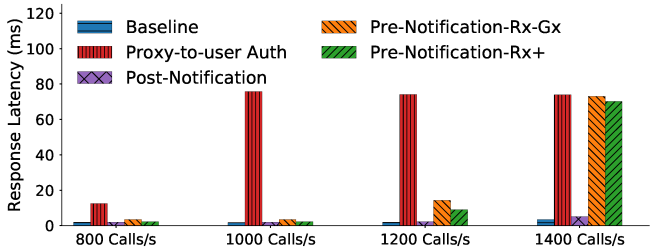


Fig. 7: Average latency in VoLTE call setup with 0% spoofed calls.

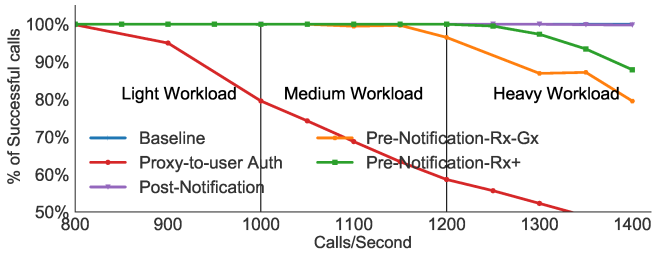


Fig. 8: Percentage of successful calls with 5% spoofed calls.

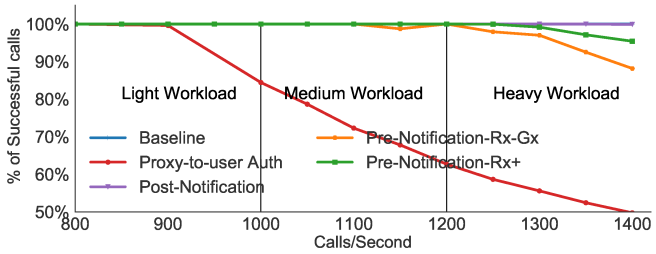


Fig. 9: Percentage of successful calls with 10% spoofed calls.

Figure 5 presents the **percentage of successful calls** with 0% spoofed calls. Proxy-to-user authentication results in significant performance degradation even under light workload. The two Pre-Notification variants do not degrade performance under light and medium workloads, but increasingly degrade performance under higher workloads. Despite utilizing additional resources at the PGW, Post-Notification results in no significant performance degradation of the overall throughput at the IMS server, even under heavy workload. The performance degradation of proxy-to-user authentication (even at light workload) is a consequence of CPU saturation at the IMS server (Kamailio).

IMS CPU utilization. Figure 6 depicts the CPU utilization of the IMS server under four different workloads. The IMS server saturates the allocated CPU core at 800 calls/second with proxy-to-user authentication. This results in severe performance degradation as the workload increases. NASCENT variants do not utilize significantly higher CPU compared to the baseline, and therefore do not result in performance degradation at light and medium workloads. At high workloads, the baseline saturates the available CPU core and therefore even Pre-Notification-Rx-Gx and Pre-Notification-Rx+ severely degrade performance.

PGW CPU utilization. Caller-ID validation solutions also require additional CPU resources at the PGW. We find that we need additional ∼15-29% CPU for the Post-Notification solution and ∼20-25% CPU for Pre-Notification-Rx+. At higher workloads, Post-Notification successfully handles a higher percentage of calls and therefore has higher CPU utilization.

Call setup latency. Since a VoLTE call is only established after a `200 OK` is received from the IMS server, any additional messages processed by IMS server will induce additional latency in the VoLTE call setup. Figure 7 presents these results. The standard deviation among the values representing each of the 10 individual runs is below 18 ms (below 4 ms for medium and light workloads) in this case. Proxy-to-user authentication incurs significant latency compared to the baseline. The three NASCENT variants do not incur high latency at light and medium workloads. At higher workloads, as evident from Figure 6, CPU saturation leads to higher induced latency with Pre-Notification-Rx-Gx and Pre-Notification-Rx+. Since Post-Notification does not introduce additional messages compared to the baseline, the latency incurred is negligible.

Results with spoofed calls. Figure 8 and Figure 9 present the results at 5% and 10% spoofed calls. Comparing Figure 8 and Figure 9 with Figure 5, we observe that the performance of caller-ID validation improves as the percentage of spoofed calls increases. For example, Pre-Notification-Rx+ results in ∼18% call loss with 0% spoofing. However, at 5% and 10% spoofed calls Pre-Notification-Rx+ results in only ∼12% and ∼4% call drop, respectively. Since Pre-Notification rejects spoofed calls before forwarding the `INVITE` message to the caller, a higher percentage of CPU is available to legitimate calls in this case.

2) NFV Deployment Model: We emulate a deployment where the EPC and IMS networks are instantiated on virtualized hardware platforms and are co-located to allow EPC and IMS VNFs to share system resources. This allows the IMS server to utilize more CPU resources and therefore we need higher workloads to saturate the IMS. Figure 10 presents the **percentage of successful calls** with 0% spoofed calls. Even with NFV deployment, proxy-to-user authentication results in significant performance degradation at light workloads. Pre-Notification-Rx-Gx also results in significant performance degradation at medium and high workloads. Since Pre-Notification-Rx-Gx relays the EPC identifiers via

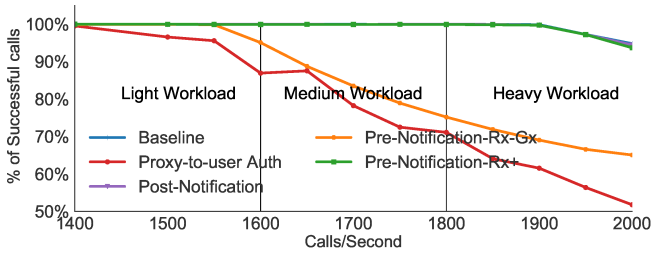


Fig. 10: Percentage of successful calls with 0% spoofed calls with NFV deployment.

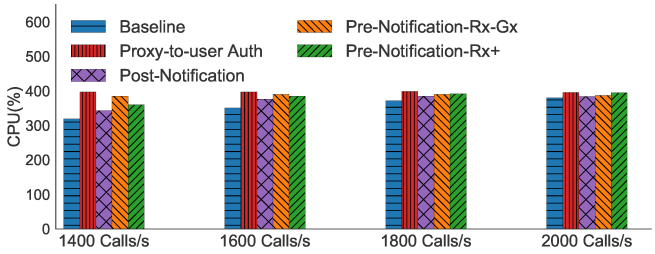


Fig. 11: CPU utilization with 0% spoofed calls with NFV deployment.

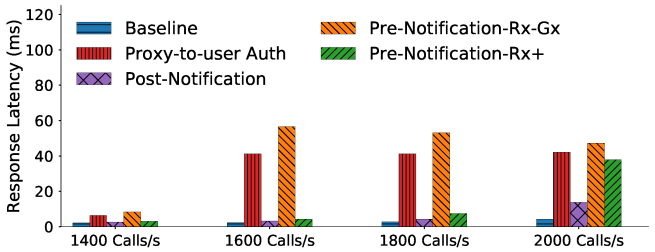


Fig. 12: Average latency in VoLTE call setup with 0% spoofed calls with NFV deployment.

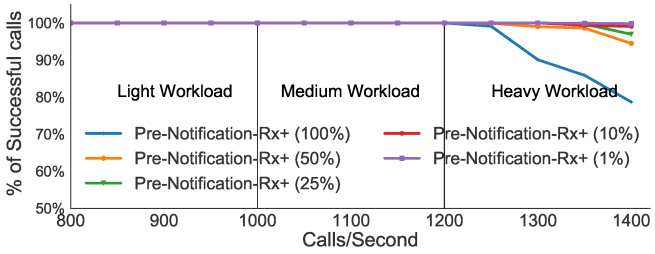


Fig. 13: Percentage of successful calls with varying spoofing percentage for the Pre-Notification-Rx+ variant. The number within parentheses indicates the % of calls validated.

the PCRF, it exhibits higher CPU utilization than the other two variants due to the additional messages processed by the PCRF.

Figure 11 presents **total CPU utilization**. Proxy-to-user authentication uses the highest overall CPU even in the absence of the PGW SIP message interception overhead. Pre-Notification-Rx+ does not exhibit significantly higher CPU utilization or performance degradation than the baseline.

Figure 12 presents the **call setup latency** incurred in NFV deployment. The standard deviation among the values representing each of the 10 individual runs is below 9 ms (below 3 ms for medium and light workloads) in this case. Pre-Notification-Rx+ incurs only negligible latency compared to the baseline and Post-Notification at light and medium workloads.

3) Selective Validation: Selective validation at the IMS can be used in cases where the network is experiencing heavy workloads. For example, IMS servers can use historical data to determine which caller-IDs to be validated. Figure 13 shows the results of Pre-Notification-Rx+ when only a specific percentage of randomly selected calls are validated. As expected, the performance impact of caller-ID validation decreases as the percentage of calls that are validated decreases. When 10% of calls are validated, Pre-Notification-Rx+ has negligible overhead.

4) Tradeoffs among the Three Variants: The three NASCENT variants offer service providers the flexibility to prioritize user experience, performance overhead, or deployment effort. Post-Notification has negligible overhead and requires no operational changes, but it may adversely impact user experience. Mobile subscribers may receive missed call notifications, and while this may be acceptable in the absence of a subscription-based (and possibly paid) caller-ID validation service, it is not ideal for end users.

Pre-Notification-Rx-Gx drops spoofed calls before the user is notified and does not require high deployment effort (no new interfaces are added), but has the highest performance overhead among the three variants. Therefore, it may not be acceptable for operators or deployments which often encounter high flash workloads.

Pre-Notification-Rx+ overcomes the shortcomings of the other two variants at the cost of higher deployment and operational effort. This variant requires standardization of a new REST based interface and requires IMS servers to directly contact the PGW serving a user. The list of PGWs currently deployed (and serving a user) can be easily configured for traditional deployments, but this is more difficult for NFV deployments where PGW and CSCFs are dynamically instantiated to meet workload requirements.

VI. Discussion

Microservice-based design. Extracting and storing subscriber identifiers can be implemented as a microservice module which can be independently deployed. Such a design has the following benefits: (a) It substantially reduces processing and storage requirements at the PGW, and (b) It allows the CSCF to directly retrieve the caller-ID from the microservice, thereby eliminating the need for PGW configuration (IP address and port) at the IMS. This design also benefits NFV-based deployments where multiple PGW instances are dynamically instantiated to handle incoming workload.

Legitimate use of caller-ID spoofing and service extensions. Caller-ID spoofing can be used in legitimate cases such as privacy protection or when a user has multiple

subscriber identifiers, e.g., preferring to show a 1-800 number [2]. NASCENT may flag these legitimate cases as caller-ID spoofing. We leave freedom to the carriers to determine what action to take once caller-ID spoofing is detected.

For instance, only spoofed calls from subscribers who use multiple or private caller-IDs, or subscribe to a legitimate spoofing service, can be allowed through. Blocking caller-ID spoofing can also be an add-on service. In NASCENT, caller-ID validation is performed at the IMS and therefore its design can be easily extended to support additional functionality. Unlike the PGW, IMS servers have access to network databases (such as HSS), which store IMS subscription information and can be used to allow legitimate caller-ID spoofing. NASCENT's mapping tables can be exposed to more services, such as SMS, to enable them to validate users.

Effective and gradual deployment. NASCENT is effective when it is deployed in the caller's network, and does not need universal deployment. NASCENT may not be helpful if only deployed in the callee's network when the forwarded ID has been spoofed. In this case, other solutions may be necessary, such as endpoint-only caller-ID spoofing detection or additional infrastructure for end-to-end authentication (e.g., via PKI or global certification infrastructure). These solutions are orthogonal and can be simultaneously used.

Extension to non-VoLTE calling. While our work focuses on VoLTE, it is conceptually applicable to other voice services such as circuit-switched calls, WiFi calling, and Internet telephony. The key idea is to enforce cross validation between the caller-ID used in the call setup and the one authenticated by the carrier networks.

Applicability to 5G. NASCENT can be naturally extended to 5G, which still uses a VoLTE-like technique to support VoIP. The use of NFV in 5G makes it even easier to detect caller-ID spoofing, as long as the proposed changes are integrated into the VNFs at the IMS and PGW. During early stages of 5G deployment, it is easier to develop built-in defense against caller-ID spoofing than to patch 4G.

VII. Conclusion

In this paper, we have proposed an effective, efficient, and easy-to-deploy solution, NASCENT, for detecting caller-ID spoofing. NASCENT performs the main cross validation operations at the IMS, hence reducing the load on the EPC dataplane gateways, but leverages authentic identifier information supplied by the EPC network. We have implemented and experimented with three variants of NASCENT, and compared them to proxy-to-user authentication. We find that NASCENT achieves its goals of effectiveness and efficiency, and the three variants offer service providers flexibility to prioritize user experience, performance overhead, or deployment effort.

Acknowledgments

This work has been sponsored in part by NSF grants CNS-1717493 and CNS-1749045.

References

[1] C.-Y. Li, G.-H. Tu, C. Peng, Z. Yuan, Y. Li, S. Lu, and X. Wang, "Insecurity of voice solution VoLTE in LTE mobile networks," in *Proc. of CCS*, 2015.
[2] FCC, "Spoofing and caller ID," https://www.fcc.gov/consumers/guides/spoofing-and-caller-id, 2018.
[3] IRS, "IRS urges public to stay alert for scam phone calls," https://www.irs.gov/newsroom/irs-urges-public-to-stay-alert-for-scam-phone-calls, 2018.
[4] TrueCaller, https://www.truecaller.com/, 2017.
[5] RoboKillerApp, "Spammed by a local call? how to stop neighbor spoofing," https://www.robokiller.com/blog/local-call/, 2017.
[6] SpoofCard, https://www.spoofcard.com/, 2018.
[7] SpoofTel, https://www.spooftel.com/, 2018.
[8] J. Song, H. Kim, and A. Gkelias, "ivisher: Real-time detection of Caller ID spoofing," *ETRI Journal*, vol. 5, no. 5, Aug 2014.
[9] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "RFC 3261: SIP: Session Initiation Protocol," 2002.
[10] J. Li, F. Faria, J. Chen, and D. Liang, "A mechanism to authenticate caller ID," in *World Conference on Information Systems and Technologies*, 2017, pp. 745–753.
[11] 3GPP, "TS 23.218: IP Multimedia (IM) session handling," 2017.
[12] B. Reaves, L. Blue, and P. Traynor, "AuthLoop: End-to-end cryptographic authentication for telephony over voice channels," in *USENIX Security*, 2016.
[13] B. Reaves, L. Blue, H. Abdullah, L. Vargas, P. Traynor, and T. Shrimpton, "AuthentiCall: Efficient identity and content authentication for phone calls," in *USENIX Security*, 2017.
[14] H. Mustafa, W. Xu, A.-R. Sadeghi, and S. Schulz, "End-to-end detection of caller ID spoofing attacks," *IEEE Transactions on Dependable and Secure Computing*, 2016.
[15] H. Mustafa, W. Xu, A. R. Sadeghi, and S. Schulz, "You can call but you can't hide: Detecting caller ID spoofing attacks," in *Proc. of DSN*, 2014, pp. 168–179.
[16] H. Deng, W. Wang, and C. Peng, "CEIVE: Combating Caller ID Spoofing on 4G Mobile Phones Via Callee-Only Inference and Verification," in *Proc. of ACM MOBICOM*, 2018.
[17] "Voice over LTE," http://www.gsma.com/technicalprojects/volte, 2015.
[18] V. Fajardo, J. Arkko, J. Loughney, and G. Zorn, "Diameter base protocol," RFC 6733, October 2012.
[19] H. Kim, D. Kim, M. Kwon, H. Han, Y. Jang, D. Han, T. Kim, and Y. Kim, "Breaking and fixing volte: Exploiting hidden data channels and mis-implementations," in *Proc. of CCS*, 2015.
[20] P. Ventuzelo, O. Le Moal, and T. Coudray, "Subscribers remote geolocation and tracking using 4g volte enabled android phone," in *Symp. on Information and Communications Security (SSTIC)*, 2017.
[21] Metaswitch, "Evaluating VoLTE security," https://www.metaswitch.com/the-switch/evaluating-volte-security.
[22] S. T. Chow, V. Choyi, and D. Vinokurov, "Caller name authentication to prevent caller identity spoofing," US Patent 9,241,013, Jan 2016.
[23] Y. Cai, "Validating caller id information to protect against caller id spoofing," US Patent 8,254,541, Aug 2012.
[24] S. A. Danis, "Systems and methods for caller id authentication, spoof detection and list based call handling," US Patent 9,060,057, Jun 2015.
[25] H. Tu, A. Doupé, Z. Zhao, and G.-J. Ahn, "Toward standardization of authenticated caller ID transmission," *IEEE Communications Standards Magazine*, vol. 1, no. 3, pp. 30–36, 2017.
[26] S. Kim, B. Koo, and H. Kim, "Abnormal VoLTE call setup between UEs," in *Proc. of International Conference on Security and Management, SAM*, 2015.
[27] 3GPP, "TS 29.212: Policy and charging control over Gx reference point," 2017.
[28] ——, "TS 29.214: Policy and charging control over Rx reference point," 2017.
[29] "Kamailio," https://www.kamailio.org/w/.
[30] 3GPP, "TS 29.213: Policy and Charging Control signalling flows and Quality of Service (QoS) parameter mapping," 2017.
[31] J. Vink, "Kore.io - an easy to use web platform for c," https://kore.io/.
[32] freeDiameter, http://www.freediameter.net/trac/, 2016.
[33] "SIPp," http://sipp.sourceforge.net/.
[34] C. Shen, E. Nahum, H. Schulzrinne, and C. P. Wright, "The impact of TLS on SIP server performance: Measurement and modeling," *IEEE/ACM Trans. Netw.*, vol. 20, no. 4, pp. 1217–1230, Aug. 2012.
[35] Docker, https://www.docker.com, 2018.