Effective and Efficient Automated Spam Call Traceback Schemes

Jianhua He, Hsiao-Hwa Chen, *Life Fellow, IEEE*, Kun Yang, Tao Gao, and Zhengwen Cao

***Abstract*—Spam calls have been a persistent issue, leading to significant economic and social harm. Spam call traceback is a crucial measure to combat spam calls by identifying fraudsters and holding problematic providers accountable. An automated spam call traceback method, i.e., Jager, has been proposed to address low efficiency issues of manual traceback. However, apart from complex cryptographic operations, it requires the traceback authority (TA) to generate a call label for every call. And all the call detail records (CDRs) are required to be stored at a central server. These generates very high and unnecessary traffic and computation loads. In this paper, we first investigate a simple automated spam call traceback method, namely distributed CDR sharing (DCS). With this method the carriers grant access of their local CDRs to an automated call traceback center (ACTC). The ACTC only accesses the CDRs for reported spam calls via secure APIs. The call path can be automatically reconstructed to locate the spam call origins. As non-cooperative carriers (such as legacy and malicious carriers) may break the traceback automation, we propose an enhanced automated spam call tracing (ASCT) method to address the issue. ASCT uses locally stored chained CDR blocks for mutual verification between carriers. Only when non-cooperative carriers are encountered, copies of CDRs are send to a central CDR server to help mitigate the impact of the non-cooperative carriers. The proposed methods are evaluated and compared to the manual and Jager methods. Experiment results show that the proposed methods are very efficient and scalable, while achieving a similar level of security performance to that of the manual method. Under the condition of all cooperative carriers, full call tracing automation can be achieved without generating any traffic to the central CDR server.**

***Index Terms*—Spam call; Spam call identification; Call authentication; Spam call traceback**

I. INTRODUCTION

Spam calls are unwanted phone calls, typically made with malicious or deceptive intent by real humans or machines (i.e., robocalls). They are often used to scam and phish for personal information or sell something there were not asked for. For example, a scammer pretending to be from a bank asks for the account details of phone users. Spam calls have been a persistent issue and are increasingly prevalent due to the rise of more advanced and sophisticated technologies. They caused significant financial losses to individual call victims and nationwide economy [1]. According to Truecaller's US Spam and Scam Report [2], about 21% of adult Americans (56 million) reported losing an average of $452 to scams, totaling over $25.4 billion. And 92% of Americans received spam calls in 2023. In 2023, 16% of UK consumers reported falling victim to phone scams.

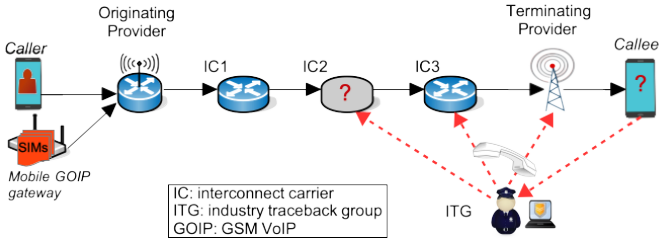

Fig. 1. Spam call traceback with manual method.

Spam calls could be classified into two categories, that is with or without call ID spoofing. For the spam calls with spoofing, the callers deliberately change the phone numbers and/or the names that are relayed as caller ID information, while the phone number and ID are not changed for the non-spoofed calls. Call spoofing is used to either hide the identity or try to mimic the number of a real company or person who has nothing to do with the real caller. These spoofed calls will trick the victims to trust the callers and give up money or sensitive personal information.

Efforts to combat spam calls with spoofing span both infrastructure-based and end-device-based technologies, each targeting different layers of the telecom ecosystem [3], [5]–[8], [11], [12]. STIR/SHAKEN (S/S) framework is example infrastructure based solution [3], which authenticates call origin through cryptographic signatures embedded in SIP signaling [3] [4]. Solutions being focused on the device side include caller reputation databases, machine learning and call back based detection methods, call back based. They monitor call behavior and block known or suspicious numbers [2], [8], [9].

For spam calls that do not involve caller ID spoofing, legitimate phone numbers are often used. These numbers may be obtained from unscrupulous internet service providers (ISPs) or registered using stolen identities. For these spam calls with real call numbers, the anti-proofing technologies such as S/S will not work. Criminals frequently manage large volumes of SIM cards to generate high volumes of spam calls. To evade detection and capture, they may employ compact

Jianhua He (email: j.he@essex.ac.uk) and Kun Yang (email: kun.yang@essex.ac.uk) are with the School of Computer Science and Electronic Engineering, Essex University, UK. Hsiao-Hwa Chen (email: hshwchen@mail.ncku.edu.tw) is with Department of Engineering Science, National Cheng Kung University, Taiwan. Tao Gao (email: gtnwpu@126.com) is with School of Data Science and Artificial Intelligence, Chang'An University, China. Zhengwen Cao (email: caozhw@nwu.edu.cn) is with School of Computer Science and Engineering, Northwest University, China.

This work was funded in part by National Science and Technology Council of Taiwan (Nos. 113-2221-E-006-143 and 113-2221-E-006-144) and by EPSRC with RC under Grant EP/Y027787/1; in part by UKRI under Grant EP/Y028317/1; and in part by the Horizon European Program under 101086228.

GSM-over-IP (GoIP) gateways, which allow them to place calls while remaining mobile and difficult to trace. In many cases, the spam call operations are controlled remotely. For instance, human operators based in foreign countries may use local software agents or devices to initiate calls that appear to come from local numbers. This distributed and deceptive setup significantly complicates real-time identification for blocking such calls.

Despite the significant impact of spam calls and ongoing efforts to combat them, there is still no universally effective solution to block such calls. A crucial complementary approach is proactive call traceback and analysis, which can help identify and prosecute fraudsters and hold problematic ISPs accountable. This not only enforces legal compliance but also serves as a deterrent to potential offenders. Currently, call traceback is mainly performed manually as illustrated in Fig. 1. In the United States, the Industry Traceback Group (ITG) was established to coordinate traceback efforts. By law service providers are required to respond promptly to traceback requests [13]. However, manual traceback is slow and inefficient. Tracing a single call can take hours or even days [13]. Given an overwhelming volume of spam calls, the number of cases that can be investigated manually is severely limited. A small team of investigators can quickly become overburdened.

To overcome the inefficiencies of manual call traceback, Adei *et al.* proposed an automated tracing method called Jager [13]. Jager is consisted of a traceback authority (TA), a record store (RS) and service providers. The service providers are required to send the call detail records (CDRs) to the RS for every call they handle. The TA is responsible for generating call labels using oblivious pseudorandom functions on requests of the providers, for every call to be stored at the RS. A designated provider is responsible for call traceback by accessing the CDRs stored at the RS. In addition, witness encryption algorithm is used by the service providers to encrypt the content of call records. With the authorization from the TA, the designated provider can access call CDRs at the RS and decrypt the cyphertexts. Furthermore, group signature technology is applied to add provider anonymity when they upload CDRs. Jager is the first reported automated call traceback method. However, apart from the complex cryptographic operations on the call records, the method requires the providers to communicate with the TA for call label generation and send all the encrypted CDRs to the RS. Such operations can lead to significant communication and computation overheads, making the TA and the RS severe bottlenecks for communication and computation.

In light of the existing challenges and research gaps in combating spam calls, we first investigate a simple automated spam call traceback method, i.e., distributed CDR sharing (DCS). It features a relatively straightforward design that relies on distributed CDR sharing by the carriers. An Automated Call Tracing Center (ACTC) can retrieve CDRs automatically via secure APIs and reconstruct the call path to locate the spam call origins. As DCS's effectiveness can be hindered by non-cooperative carriers that refuse to share their records, we proposed a more resilient and efficient automated spam call traceback (ASCT) method. ASCT utilizes chained CDR blocks and a central CDR server to ensure traceability even in the presence of uncooperative carriers. Carriers on the call path cryptographically sign and verify blocks of CDRs, ensuring their integrity and authenticity. CDRs are stored locally at the carriers. A copy of CDR is sent to the central CDR server only when non-cooperative carriers are encountered. ACTC initiates traceback from the destination carrier, sequentially retrieving CDRs from upstream carriers. These records are analyzed to reconstruct the complete call path and identify the spam call origin. The resulting trace can then serve as a verifiable evidence for prosecution.

The contributions of the paper can be summarizd as follows. Firstly, we propose a simple automated spam call traceback method DCS, which stores call records locally at the carriers. It can avoid the communication and computation issues of fully storing the CDRs at a central server in Jager, while achieving faster call tracing than and equivalent security performance as the manual method. Secondly, we propose an enhance automated traceback method ASCT to address the DCS issue of tracing automation breaking down with non-cooperative carriers. ASCT uses a hybrid way of storing CDRs. On top of locally stored CDRs, a copy of CDR is sent to the central CDR server only when a non-cooperative carrier is encountered. ASCT can greatly reduce the communication traffic and computation loads compared to Jager. Thirdly, a new framework is proposed to evaluate the spam call traceback schemes with performance metrics of security performance, computation and communication overheads, and manual check loads. Experimental results demonstrate that effectiveness and efficiency of the proposed methods, and a comparable level of security to that of the manual approach. Notably, if all carriers cooperate, full automation can be achieved without incurring additional traffic to the central CDR server, while Jager will need to send every CDR for all calls to the server.

The rest of the paper is organized as follows. The SIP limitations and representative spam call measures are discussed in Section II. Representative solutions against call spoofing are presented in Section III The assumptions and objectives of the automated traceback system are presented in Section IV. The design of automated call tracing methods is presented in Section V. Performance analysis and discussions are presented in Section VI. Section VII concludes the paper.

II. SIP LIMITATIONS AND REPRESENTATIVE SPAM CALL MEASURES

As the majority of Internet-based calls and spam calls involve SIP, we begin this section with a brief introduction to the protocol [14], [15]. Then, we discuss its limitations and outline common spam call techniques. While this study focuses primarily on tracing SIP-based spam calls, the proposed methods are also applicable to other forms of spam call traceback

A. Session Initiation Protocol and Call Detail Records

SIP is a signaling protocol used to initiate, manage, and terminate real-time voice, video, and messaging

communication sessions over IP networks. It has been widely used in Voice over IP (VoIP) services, allowing endpoints such as softphones, mobile VoIP apps, and SIP-enabled PBXs to establish call sessions. A typical SIP call begins when the caller's device (i.e., User Agent Client) sends an INVITE request to the callee through a series of SIP servers or proxies. This request traverses intermediate entities like SIP proxies, Session Border Controllers (SBCs), or interconnect carriers, each of which may add routing or policy enforcement headers such as Via, Record-Route, or Call-ID. Once the callee accepts the call (via a 200 OK response), the caller confirms with an ACK and the media (voice/video) session begins, typically using RTP.

For billing and call tracking purposes, CDRs are generated by SIP elements such as SIP servers, proxies, and gateways. These records contain metadata like call start/end times, duration, caller/callee identifiers (SIP URIs or phone numbers), route taken, codecs used, and call status. Interconnect carriers (ICs) and VoIP service providers rely on these records for usage-based billing, fraud detection, and compliance audits.

B. SIP Limitations

While SIP is foundational to modern IP-based voice communications, it also presents several technical and operational challenges when it comes to combating spam, scam, and robocalls [14]. These challenges arise largely from the protocol's inherent flexibility, decentralized architecture, and trust-based assumptions.

1) Lack of Strong Authentication: One of SIP's major weaknesses is its lack of built-in identity verification. SIP messages, such as the INVITE request, allow the caller to specify the "From" head, which is often interpreted as the caller ID. However, SIP does not require cryptographic verification of this header, meaning that caller ID spoofing is trivial. Spammers may exploit this to impersonate trusted entities (such as banks, government agencies, or local numbers), making it harder for users to distinguish real calls from fraudulent ones.

2) Header Manipulation and Intermediary Stripping: As SIP messages traverse multiple intermediaries (such as SIP proxies, SBCs, and interconnect carriers), various headers can be modified, stripped, or rewritten. This is often done for privacy, topology hiding, or interoperability. However, it also introduces barriers to traceback, as critical headers like Via, Route, P-Asserted-Identity, or Call-ID might be altered or removed. As a result, downstream carriers or enforcement entities may not be able to trace a call's origin or path accurately, especially when SIP elements are misconfigured or intentionally malicious.

3) Decentralized and Borderless Architecture: SIP was designed for interoperability across domains, which is a strength for communications but a weakness for accountability. Spammers can originate calls from jurisdictions with loose regulations, routing them through grey routes and unregulated gateways. Carriers operating in different countries may not share consistent security policies, CDR formats, or traceback mechanisms. This cross-border complexity hinders law enforcement and regulatory actions, even when abuse is detected.

4) Lack of End-to-End Visibility: In SIP networks, especially with Least-Cost Routing (LCR), calls may traverse multiple untrusted hops, each of which may repackage or mask the signaling details. Intermediate carriers (ICs) may have no knowledge of the original caller, and vice versa. The absence of global traceback standards makes it difficult to reconstruct the full path of a suspicious call after being detected.

5) Inadequate Incentives and Capabilities for Filtering: Unlike emails, where spam filters are deployed extensively, many SIP providers lack robust spam detection systems. Carriers may be reluctant to invest in real-time SIP analysis due to costs or may not have incentives if they profit from call volume. SIP's real-time nature also makes deep packet inspection or heuristic analysis difficult without impacting quality of service.

C. Representative Spam Call Techniques

While SIP is vital for modern voice communications, its security and accountability limitations make it vulnerable to spam and scam callers. Spam and scam callers employ a variety of technical methods to evade detection, disguise their identity, and maximize reach. One of the most prevalent issues is caller ID spoofing as discussed in Section I. Another major problem involves the use of GoIP gateways, devices that allow VoIP calls to be routed through SIM cards from real mobile carriers. Spammers can insert legitimate SIMs into these gateways to originate calls using real phone numbers, including the numbers from trusted operators. This makes detection difficult, as the caller ID corresponds to a valid account. Spammers can also exploit MVNOs (Mobile Virtual Network Operators), allowing them to register a large number of SIM cards with minimal identity verification. These SIMs can then be used for mass calling or resold to other malicious actors. Additionally, grey routes (illegitimate or unregulated carrier interconnects) can be leveraged to bypass scrutiny and reduce call delivery costs. Least-cost routing (LCR) systems are often abused to inject spam traffic into vulnerable networks across multiple intermediaries, obscuring the origins. These methods are often combined and rapidly rotated to stay ahead of detection systems. Combating them requires improvements in call authentication (e.g., STIR/SHAKEN), SIM provisioning oversight, and carrier-level traceback cooperation.

III. Existing Solutions to Combat Spam Calls

In Section I we have briefly introduced the solutions for call ID spoofing and call traceback. In this section we present more details on the solutions for call ID spoofing.

S/S is one of the most influential anti-call ID spoofing methods. The service providers in the United States are required to implement the S/S. S/S was mainly developed by industry and adopted for use in the United States [3]. The STIR part representing a suite of protocols developed by an Internet Engineering Task Force (IETF) working group STIR, while the SHAKEN standard was jointly developed

by ATIS and session initiation protocol (SIP) Forum. It was designed to manage the deployment of secure telephone identity technologies to provide end to end cryptographic authentication and verification of the telephone identity in VoIP networks. However, its effectiveness is limited in scenarios involving legacy PSTN systems, international routing through non-compliant networks, or spoofed SIMs. Majority of calls arrive without a signature, which significantly limit the effectiveness of S/S in practice. The AB Handshake protocol is a complementary measure to mitigate this issue, which verifies the legitimacy of calls between originating and terminating providers using secure API exchanges [5]. While it offers stronger end-to-end validation, it depends on broad adoption and bilateral cooperation between carriers.

Some real-world and practical standalone applications have been used to block spam calls. For example, the victims of spam calls users and telephone operators may report malicious phone numbers, which may be collected by government security department or security companies (e.g., Truealler) and included in blacklists. These blacklists are shared with app users. Malicious calls with numbers included in the blacklists will be blocked. These approaches are simple and offer immediate user protection. But they have a major limitation that they are not effective for calls with numbers not in the blacklists. Spammers can use new phone numbers, which can be easily obtained. And they may check the blacklists and spoof with other non-reported numbers.

The UK government recommended mobile phone users to hang up suspicious calls and manually call back. To address the potential issues of STIR/SHAKEN, Feng *et al.* proposed a Spoof Against Spoofing (SAS) approach [12], in which the caller mobile phone used spoofing to combat spoofing by requesting the mobile phones with the calling number to call back. While the SAS approach may help prevent call spoofing, it may introduce new security risks due to installation of new app at phones and not work when infrastructure based solution is in place which will stop call spoofing. Furthermore, there is no efficient way of announcing if a mobile phone supports SAS. The phones implementing SAS may be subject to security attacks as they could switch off the SAS function after being exhausted by a large number of requests from malicious users to call back and then ignore the requests from legitimate users.

IV. ASSUMPTIONS AND DESIGN OBJECTIVES

For the automated spam call traceback system, we assume a heterogeneous and partially cooperative telecom environment. It consists of SIP-based carriers, legacy PSTN gateways, VoIP providers, mobile network operators (MNOs), and ICs. Each participating entity is expected to generate and store local CDRs and basic metadata logs (e.g., SIP headers, timestamps, IPs, media paths) in an almost real-time manner. While core telecom and major Tier-1 carriers may support standard SIP tracing (e.g., via S/S or AB Handshake), the system accounts for legacy or grey-route providers who may strip, modify, or withhold SIP header fields (such as Via, From, or Call-ID). The assumption of cooperative carriers is based on adoption of new laws like the TRACED Act and other changes to regulations in the United States. Network operators, at the mandate of regulators and legislators, are required to implement two schemes to address the spam calls [13]. The first is the use of an ITG for spam call tracing back, which serve as a central point of contact between traceback seekers and network operators. All carriers are mandated to respond to traceback requests within a 24 hour period. The second scheme is a requirement for all VoIP operators to support the call authentication mechanism S/S. Therefore, we believe the assumption of a proportional cooperative carriers is reasonable.

In a typical SIP call, especially over the public Internet or across multiple carriers, it is assumed that each intermediary only sees the hop immediately before and after it [13]. There is no shared global ID tying together each carrier's internal records. Malicious or lax carriers can break or drop logging and killing traceability. So if spam calls route through $4\sim5$ carriers, even if the mobile operator at the destination wants to trace it back, they often hit a dead end due to header rewriting or stripping.

Additionally, malicious or non-cooperative IC may be present, especially in low-cost or international routing chains. These carriers may manipulate routing paths or spoof identities, potentially breaking header continuity. Despite these threats, it is assumed that a sufficient portion of the call path, especially the origin or destination segments, is cooperative. And these carriers log internally unique traceable identifiers, such as home location register (HLR), International Mobile Subscriber Identity (IMSI), IP address, and trunk information. In spam call traceback, these identifiers play crucial roles in reconstructing the call's origin. The HLR, maintained by mobile operators, holds subscriber profiles and network locations, enabling investigators to trace which operator and region a suspicious mobile number belongs to. The IMSI, stored on the SIM card, uniquely identifies the user on the mobile network and is useful for linking calls to specific SIMs, even if caller IDs are spoofed. IP addresses, recorded by SIP servers and SBCs during VoIP calls, can reveal the originating device or network, especially when tracing calls from internet-based sources. Trunk information, including trunk group IDs and carrier routing codes, helps map how a call was handed off between networks, identifying specific ICs or gateways. Together, these data points enable correlation across networks to track the path of spam calls and isolate tampered or spoofed segments.

The core objective of automated traceback methods is to find the original provider or ICs that misconduct or a spam call come from with an automation level as high as possible. It is achieved through redundant metadata correlation, cryptographic tagging (e.g., Identity headers), and inter-provider API-based queries, without reliance on full trust across all intermediaries. Additionally, the automated traceback methods should achieve a traceback performance comparable to or higher than manual traceback method in terms of success rate and evidential integrity, without introducing extra system security risks.

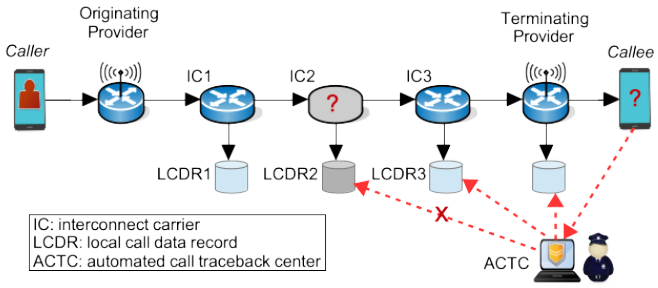

Fig. 2. Distributed CDR sharing method for automated spam call traceback.

V. DESIGN OF AUTOMATED CALL TRACING METHODS

In this section, we first investigate a simple and intuitive automated call tracing method DCS based on distributed CDR sharing with the ACTC. Then we will propose the enhanced ASCT with hybrid CDR storage to address the limitations of the DCS method.

A. Distributed CDR Sharing Approach

The DCS method had a relative simpler design with a collaborative mechanism. The telecom providers shared their CDRs with the ACTC, and the ACTC would synchronize call metadata to reconstruct the full path of a call across multiple networks to help automate call tracing. An illustration of DCS method is presented in Fig. 2.

With this method each provider (originating, ICs, and terminating) logs CDRs at local CDR (LCDR) servers, which contain per-hop log details, such as Call-ID, timestamps, caller/callee info, and SIP headers. By aligning and matching these CDRs using cryptographic call tokens (e.g., S/S Identity headers) or consistent metadata (e.g., timestamp and called number), a unified end-to-end view of the call path could be assembled. To prevent the disclosure of sensitive CDR information to potential cyberattacks, the CDRs can be encrypted using a private key shared by each carrier and the ITG or the public key of the ACTC. This approach is especially critical for spam call traceback, where calls often pass through multiple domains, including potentially untrusted or international ICs.

To enable automated traceback, call metadata sharing APIs (standardized or bilateral) are needed. If a destination carrier or a callee flags a spam call, the ACTC can be notified (possibly via ITG) and start the traceback process to construct the call path. It is assumed that the CDRs are indexed by the hash of the called number, which is expected to be the same for all the CDRs of the carriers in the path of an investigated call. As illustrated inFig. 2, the ACTC can retrieve the CDRs related to the investigated spam call hop by hop, starting from the destination carrier. It will firstly be authenticated to access the database of the carriers using some user authentication (UA) schemes. After authencation it can query the metadata database via secure APIs with the hash index of the called number and the call timestamp. After the ACTC receives the CDR from the destination carrier, it extracts the information of the upstream carrier (IC3 in Fig. 2) and queries the upstream carrier via metadata API. The upstream carriers return the matching CDR. The ACTC then verifies the signature and timestamp of the returned CDR. Tampering can be detected when inconsistency is found in key fields (such as caller ID or originating number) between the CDRs of two adjacent carriers. This process continues recursively along upstream until either the originating carrier is identified or tampering is detected. If no tampering is found at the end of the process, the originating carrier is deemed accountable for the spam call.

The main benefits of this DCS approach include increased visibility, improved traceback speed, and ensured accountability across heterogeneous networks. It helps overcome issues like spoofed headers or stripped metadata, as correlation relies on independently logged records rather than the trust in signaling alone. It is a promising framework for scalable, automated, and cross-border spam call mitigation. However, the DCS method has a major limitation. Cooperation among providers is uneven. Some may lack the infrastructure or policies to participate, while others (especially legacy networks) may not support automated CDR sharing at all. The automated traceback process breaks down entirely if even a single IC refuses to cooperate. In some cases, malicious ICs may deliberately tamper with or falsify CDR records and deny any involvement in spam traffic. For instance, as shown in Fig. 2, if a spammer routes a call through IC2 (a grey route), and finally through IC3 to the mobile operator, the traceback may fail if IC2 strips headers and withholds metadata. Without a shared trust framework or verified linkage between the other ICs and the mobile operator, the call path cannot be reconstructed, which will result in accountability failure.

B. An Enhanced Automated Spam Call Tracing Method

To address the potential issues of non-cooperation among carriers and CDR tampering by malicious ICs, we propose an enhanced automated spam call tracing method ASCT. ASCT uses hybrid CDR storage and mutual hop wise CDR verification between neighbor carriers. An illustration of the ASCT method is presented in Fig. 3.

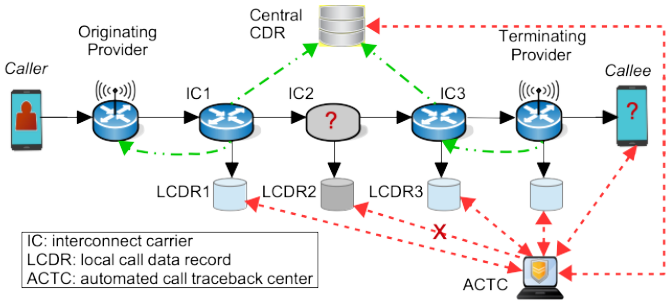

Fig. 3. Enhanced automated spam call traceback method ASCT with hybrid CDR storage.

1) Mutual Hop Wise CDR Verification: As done with the DCS method, the participating carriers (including originating providers, ICs and terminating providers) will log CDRs at LCDR servers. Additionally, the participating carriers are

required to implement a mutual hop wise CDR verification for the key fields of CDRs, which is used to protect from denial of CDR tampering. The first carrier (i.e., origination provider) will sign on the key fields of its CDR and forward the signature alongside the key fields to the downstream IC for a given call. The information can be forwarded via standard interfaces (such as SIP INVITE signaling message) or bilateral reliable data transport interfaces. The downstream carrier (i.e., IC1 in Fig. 3) will acknowledge the receipt of the signed CDR key fields with its own signature over the signed key fields. Again, the signed acknowledgement can be sent via standard interface (such as SIP ACK message used to confirm the call) or bilateral reliable transport interface. Under the ideal conditions without non-cooperative carriers, the mutual verification process will be repeated until the destination provider is reached.

The process can be illustrated in Fig. 4 with the data organized in chained data blocks. In Fig. 4, the data blocks include the signatures of the carriers and the preset key fields of CDRs corresponding to the processed call. These blocks are chained through the CDRs, which could be hashed to preserve privacy. It is noted that the notations *E1* to *E9* in Fig. 4 represent the events with a chronological order in the transmission of data blocks.

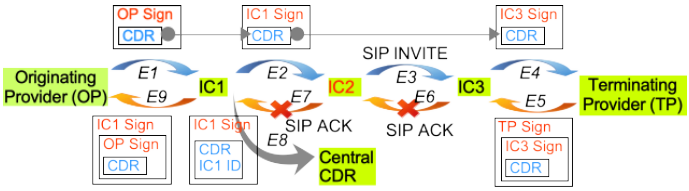

Fig. 4. Visualization of the chained CDR blocks in the ASCT method.

2) Dealing with Non-cooperative Carriers: In this paper non-cooperative carriers for automated traceback methods means that these carriers don't support the corresponding methods (such as DCS and ASCT), or intentionally don't follow the operation of these methods on sharing the CDRs. For example, the legacy PSTN carriers may not implement and support the automated traceback methods. And malicious carriers may have implemented the automated traceback methods but they may not follow the operation of these methods. If there are non-cooperative carriers, the mutual hop wise verification process cannot be completed. To address this issue, a central CDR (CCDR) server is added to store signed CDRs from some carriers when they encounter non-cooperative carriers. In the data block sent to the server, it contains extra information of the IC1 ID. It can be encrypted using the public key of the ACTC to avoid others obtaining the owner information of the deposited data blocks sent to the central CDR server. Suppose that an IC (e.g. IC1 in Fig. 3) sends a signed CDR to a non-cooperative downstream carrier (i.e., IC2 in Fig. 3). IC2 does not return an acknowledgement of the receipt of the signed CDR from IC1. Under the condition that an acknowledgement is not received from the downstream carrier, IC1 needs to sign a block of data (with the callee number and IC1 carrier ID) using its private key, and send the signed block to the central CDR. The CCDR server will generate an index for the signed block using the callee number and a timestamp for the block using the receipt time. UA can be implemented to control the CDR uploading to the CDR server.

It is possible that some of the carriers are partially cooperative, which means that the API for CDR sharing is available at some carriers but they share only part of their CDRs. The ACTC could detect such partial cooperation if it is not able to retrieve any CDR that is expected to be available at some carriers. In this case the partially cooperative carriers can be treated as non-cooperative carriers. And the ACTC can punish these partially cooperative carriers and requires them to be fully cooperative.

It is noted that ACTC and CCDR server are separate components. They have different roles in the spam call traceback systems. ACTC is the central entity executing the tasks of tracing spam calls. It exists in both manual and automated spam call tracing systems. On the other hand, the CCDR server is proposed to be used for only the ASCT scheme to store CDRs. It is not included in the manual and DCS based call tracing systems.

3) Automated Traceback Process: If a spam call is flagged by a mobile user or carrier, the ACTC will be notified and start a traceback process. It will query the metadata APIs to retrieve the CDRs from the LCDR servers using the index generated from the hashed callee number and the call start time. When it encounters a non-cooperative IC (e.g., IC2 in Fig. 3), which does not return a local CDR, the ACTC will perform UA with the CCDR server and query the CDR database with the CDR index and the call start time. If a signed block with matched CDR index and call start time is returned, the ACTC can extract the IDs of the carriers which generated the signed block and query the metadata API of the corresponding carriers (i.e., IC1 in Fig. 3). The traceback process can then be resumed with the newly retrieved local CDRs. The above process is repeated until the destination provider is reached or tampering is detected.

If no tampering is detected, the ACTC will analyze the reconstructed call path, which may be complete or partial (with non-cooperative carriers). If the call path is complete, the originating provider is deemed accountable. If the call path is partial, the available information from the retrieved CDRs will be used to detect any inconsistence in the key call information (such as caller and callee numbers, and call ID). If no inconsistence is detected, again the originating provider is deemed accountable for the spam call. Otherwise, the ACTC will need to investigate the case manually by contacting the related carriers. For example, if there is inconsistence between the key fields of CDRs from IC1 and IC3, the non-cooperative carrier(s) (i.e., IC2 in this example) can be identified from the retrieved CDRs. These carriers are most suspicious and will be investigated by the ACTC manually on the spam call issue. If needed, the related carriers (such as IC1 and IC3 in Fig. 3) will also be contacted to find out a clear cause of the investigated spam call. The non-cooperative carriers can be forced to comply with the traceback requirements and may be blacklisted if they do not cooperate on the automated call tracing.

VI. PERFORMANCE ANALYSIS

In this section, we evaluate and analyze the performance of the proposed automated traceback methods (DCS and ASCT) in terms of security performance, computation complexity, communication overhead and manual check loads. They are compared to two other traceback methods, the fully manual one (denoted by *Manual*) and the automated method Jager [13].

A. Security Performance

The core security objective of the traceback methods is to recover the full call path and identify the originating carrier of spam calls (i.e., the originating provider or malicious carrier) in the call path. The manual approach is a baseline, which can fully recover the spam call path and identify the spam call origins with a very high cost of manual checks and large delay. DCS and ASCT are more efficient, which can achieve the core security objective as the manual approach, but with much less and even no manual check under some conditions. They can significantly reduce the expenses and latency of the manual checks. The fast automated tracing method DCS and ASCT hold great potentials for secure and real-time spam call traceback. The Jager method, as discussed in Section 1, uses complex cryptographic operations for storing the CDRs at the RS. It is believed that for the Jager method the authorized providers to investigate spam calls can extract the CDRs from the RS and recover the full path of spam calls with additional manual investigation on some of the non-cooperative carriers.

Another key security objective of traceback methods is the preservation of sensitive call information of the carriers. They should minimize the risk of sensitive information disclosure due to the access and analysis of the CDRs. In the manual approach, the ITG officers contact all the carriers in the path of spam calls, requesting CDRs and analyze the CDRs to identify the spam call origins. Certain user authentication schemes will be needed to get access to the local CDRs from the carriers. For the Jager method, the providers are given the role of tracing spam calls. These providers may get access to a large volume of CDRs and sensitive call information, which should be avoided in practice. For the DCS and ASCT methods, the CDRs stored at the local servers of the carriers can be accessed by the ACTC with similar user authentication schemes to that used in the manual approach. To prevent the ACTC to access CDRs of calls that are not to be traced as requested by the ITG, the ITG can authorize the ACTC to access only the CDRs of calls to be traced with a signature of the ITG. That signature signs on the call number of traced spam calls. The ACTC presents the signed call numbers to the carriers to access the needed CDRs. Therefore, the local CDRs can be protected as in the manual approach. For the CDRs stored at the CCDR server, they are encrypted with the ACTC's public key and their call labels (called numbers) are hashed. And the ACTC also needs to be authenticated by the CCDR server and present a signature of the ITG signing on the hashed call numbers. Then the data stored at the CCDR server can be safely protected for the ASCT method. Furthermore, the data stored at the CCDR server can only include the carrier IDs of the current hop (corresponding to the reporting carrier), the upstream and downstream carriers. The ACTC can use the hop information to retrieve the complete CDRs from the reporting carrier or contact its neighbor carriers. In this way the risk of CDR leakage at the CCDR server can be mitigated. According to the above analysis, the data security performance of the DCS and ASCT are equivalent to that of the manual approach.

Next we present a brief analysis of operation complexity of the automated traceback methods in term of time. The complexity analysis will focus the central servers as they are more likely to be system bottlenecks. We consider mainly the cryptographic operations of UA (or digital signature), encryption (decryption) and hashing. Other operations can be included in the computation as well if needed. Let T_{ua}, T_{enc} and T_{hash} denote the time for the above cryptograph operations, respectively. We assume N_{ca} carriers and a proportion P_{cop} of these carriers being cooperative. We assume N_{call} calls per second. And the probability of a call to be traced is P_{tr}.

In the DCS method, the ITG plays a light role of assigning traceback tasks to the ACTC, which does not have much computational load. There is no CCDR server in the DCS system. The ACTC will only trace spam calls which are assigned by the ITG. For a general cooperative carrier, the ACTC will perform one UA to get access to the local encrypted CDR which is stored at the carrier, and one decryption to get the plaintext CDRs. Let T_{dcs} denote the cryptographic operation related time for all traced spam calls with DCS. It does not count the time spent on the non-cooperative carrier, which needs manual check and will be analysed in the next subsection. We can approximate T_{dcs} by $T_{\text{dcs}} = N_{\text{call}} P_{\text{tr}} N_{\text{ca}} P_{\text{cop}} (T_{\text{ua}} + T_{\text{enc}})$.

For the ASCT method, there are two potentially busy servers, ACTC and CDR server. Let $T_{\text{asct, actc}}$ and $T_{\text{asct, cdr}}$ denote the cryptographic operation time of ACTC and CDR server for all calls with ASCT, respectively. The ACTC time can be computed with the time to get a local CDR from each cooperative carrier and the time to get a CDR stored at the CDR server for each non-cooperative carrier. It is noted that the above calculation is not accurate but an approximation for the comparison. To get a CDR copy from the CDR server, the ACTC will go through one hashing, one UA and one decryption operation. We can have

$$T_{\text{asct, actc}} = N_{\text{call}} P_{\text{tr}} N_{\text{ca}} [P_{\text{cop}} (T_{\text{ua}} + T_{\text{enc}}) + (1 - P_{\text{cop}})(T_{\text{ua}} + T_{\text{enc}} + T_{\text{hash}})].$$

The CDR server will perform one UA for each cooperative carrier to upload a CDR when it encounters a non-cooperative carrier, and perform one UA for the ACTC to access each CDR of traced spam calls. We can approximately computed $T_{\text{asct, cdr}}$ by:

$$T_{\text{asct, cdr}} = N_{\text{call}} N_{\text{ca}} (1 - P_{\text{cop}}) T_{\text{ua}} (1 + P_{\text{tr}}).$$

For the Jager method, there are three main servers, TA, RS and ACTC (a designated provider). Let $T_{\text{jag, ta}}$, $T_{\text{jag, rs}}$ and $T_{\text{jag, actc}}$ denote the cryptographic operation time of the TA, RS and ACTC for all calls with Jager, respectively.

The main cryptographic operations of the TA include one UA with each cooperative carrier and one encryption for

each call on the generation of call labels, and one UA with the ACTC and one encryption for each traced call on the generation of call labels. We can compute $T_{\text{jag, ta}}$ by:

$$T_{\text{jag, ta}} = N_{\text{call}} N_{\text{ca}} P_{\text{cop}} (1 + P_{\text{tr}})(T_{\text{ua}} + T_{\text{enc}})$$

The main cryptographic operations of the RA include one UA with each cooperative carrier to upload a CDR for each call, and one UA with the ACTC to access the CDR of the traced calls for each cooperative carrier.

$$T_{\text{jag, rs}} = N_{\text{call}} N_{\text{ca}} P_{\text{cop}} T_{\text{ua}} (1 + P_{\text{tr}}).$$

For the ACTC its main cryptographic operations include one UA with the TA and one encryption for each cooperative carrier to get the call label of each traced call, and one UA with the RS and one decryption to obtain the CDR of the traced calls for each cooperative carrier. We can compute $T_{\text{jag, actc}}$ by:

$$T_{\text{jag, actc}} = 2 N_{\text{call}} P_{\text{tr}} N_{\text{ca}} P_{\text{cop}} (T_{\text{ua}} + T_{\text{enc}}).$$

For a quick numerical comparison, we follow the settings of [13] with $N_{\text{call}} = 10000$, $T_{\text{ua}} = 0.419ms$, $T_{\text{enc}} = 0.847ms$, and use customized settings of $T_{\text{hash}} = 0.02ms$, $N_{\text{ca}} = 5$, $P_{\text{cop}} = 0.9$, $P_{\text{tr}} = 0.01$. Under the above settings, we obtain T_{dcs}=569.7 ms, $T_{\text{asct, actc}}$=638 ms, $T_{\text{asct, cdr}}$=2116 ms, $T_{\text{jag, ta}}$=57540 ms, $T_{\text{jag, rs}}$=19044 ms, and $T_{\text{jag, actc}}$=1139 ms. It can be observed that the ACTC of the DCS, ASCT and Jager methods has relatively low computation load. The CDR server of the ASCT has less than 3 times higher computation load than that of the ACTC of DCS method. On the other hand, the Jager TA and RS have much higher computation loads, which are about 100 times and 33 times of the DCS ACTC. The TA and RS could become bottleneck of the Jager system. The result of the RS is in line with that reported in [13]. Their evaluation showed that the RS can verify only 432 submissions per core per second. The above results show that the proposed methods DCS and ASCT are much faster and more scalable.

B. Communication Load and Manual Check Workload

In addition to security performance, the efficiency of communications and the reduction of manual verification workload are also critical to the success of traceback methods.

We implemented a system level simulator in Matlab to quantitatively evaluate and compare the performance of the four traceback methods. In the simulations, the number of carriers in the call paths (including the originating and destination providers and ICs) varies from 4 to 10 with a step of one. This setting is in line with an ITG report that tracebacks usually go through 4 or more hops [13]. There carriers are classified to normal (cooperative) carriers, legacy carriers, and malicious carriers. Legacy carriers and malicious carriers are assumed to be non-cooperative, which do not share CDRs via APIs with the ACTC, but they will respond to manual CDR request from the ITG and ACTC. Moreover, malicious carriers are thought to spoof calls (e.g., change the caller numbers) and may tamper their CDRs. The proportion of legacy carriers (denoted by P_{leg}) is set to 0, 0.25 and 0.5, respectively. The proportion of malicious carriers (denoted by P_{mal}) is set to 0 and 0.1, respectively. Each mean value is obtained via 1,000 simulations. For each simulation, a spam call is generated so we can focus on the simulation of tracing spam calls by the ACTC.

The communication load is computed as the average number of CDR packets sent to the central CDR server from the carriers in one call path. The traffic between the adjacent carriers is not counted as the signatures for the CDRs could be sent via the SIP signaling messages, which will not add much additional traffic load. The packets sent from the normal carriers to the CCDR server could cause communication problems, making it a communication bottleneck. As the manual and DCS methods do not use the CCDR server, their communication loads to CCDR are zero. The mean traffic volumes of the ASCT and Jager methods are presented in the Fig. 5. For results in the upper and lower parts of Fig. 5, the proportion of malicious carriers P_{mal} is 0 and 0.1, respectively. It is observed that as Jager requires every cooperative carrier sending a CDR to the CCDR server, the traffic to CCDR server increases linearly with the number of cooperative carriers. Under the condition that all the carriers are cooperative (i.e., $P_{leg} = 0$ and P_{leg}=0), there is no traffic to the CCDR server with the ASCT method, while the traffic is the highest with the Jager method. With an increasing proportion of legacy carriers, the traffic to the CCDR server with the Jager method drops accordingly. The traffic to the CCDR server with the ASCT method increases but is still much less than the Jager traffic. The results with $P_{mal} = 0.1$ (as shown in the lower part of Fig. 5) show a similar trend as observed with $P_{mal} = 0$.

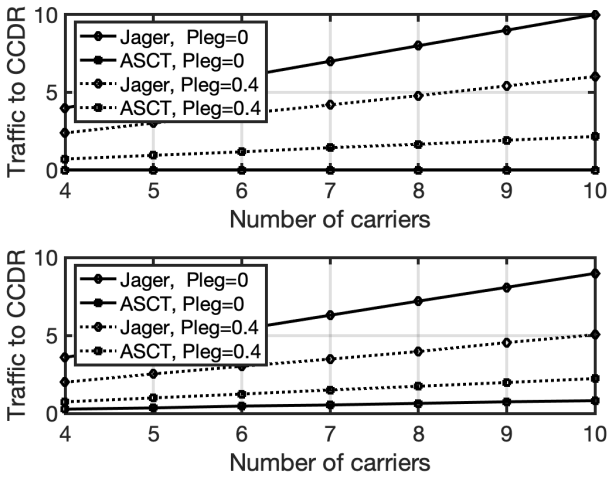

Fig. 5. Mean traffic load to CCDR server for the Jager and ASCT systems, measured in the number of CDR packets. Upper subfigure: $P_{mal} = 0$; Lower subfigure: $P_{mal} = 0.1$.

The results of manual check workload are presented in Fig. 6 for the manual, ASCT and DCS methods, which is measured by the average number of carriers manually checked by the ACTC for a spam call. As the call path reconstructions in the Jager and ASCT methods are similar, the manual check results of the Jager method are not presented. Again, the proportion of malicious carriers P_{mal} is 0 and 0.1, for the results presented in the upper and lower parts of Fig. 6,

respectively. For the results without malicious carriers (i.e., upper part of Fig. 6), the number of manual checks for spam calls is zero with the ASCT method. The ASCT method shows a superior performance in terms of communication and manual check loads. Even with malicious carriers (as shown in the results in lower part of Fig. 6), the manual check load of ASCT method increases slightly, much lower than the DCS and manual methods. Without surprise the manual method has the highest manual check load, as ACTC needs to investigate all the carriers in the call path. On the other hand, the DCS method has a middle level of manual check load. If all the carriers in the call path are cooperative, the DCS's manual check load is zero. But its manual check load increases largely with the proportion of non-cooperative carriers. Without use of the CCDR server, the DCS method is simpler and potentially securer than the ASCT method, at the costs of more manual checks (leading to a much longer traceback delay). Therefore, there is a trade off on the security, automated traceback efficiency, and speed for the two proposed methods DCS and ASCT.

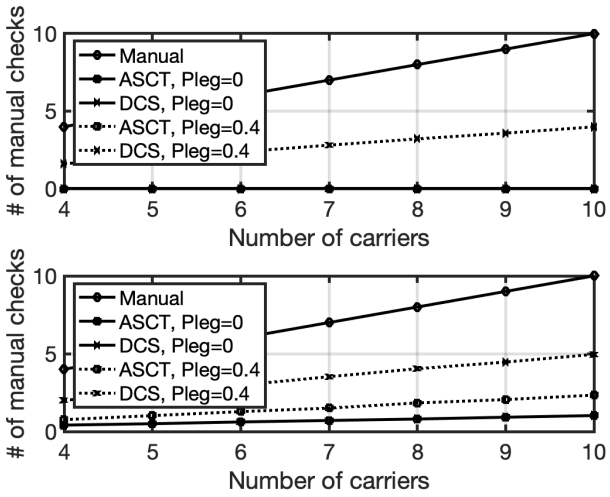

Fig. 6. Mean manual check loads for the manual, ASCT and DCS methods, measured in the number of carriers checked manually for a call. Upper subfigure: $P_{mal} = 0$; Lower subfigure: $P_{mal} = 0.1$.

VII. CONCLUSION

Spam calls have significant adversary economic and social impacts and the progress to stopping them is slow with no practical effective solution. In this paper, we investigated spam call traceback, which can complement with the spam call blocking measures and deter the potential malicious providers and spam call criminals. We first analyzed the spam call tracking techniques and studied a simple automated call tracing method DCS, which is based on distributed CDR sharing. To address the limitations of the DCS method, an enhanced spam call traceback method ASCT was proposed. ASCT uses hybrid CDR storage and mutually hop wise CDR verification. The proposed methods were evaluated by simulations and compared to the manual call traceback and an existing automated tracing method Jager. Experiment results demonstrated the effectiveness and efficiency of the proposed methods in terms of communication and computation overheads and manual investigation efforts. Under the condition of non-malicious carriers, the ASCT method can identify the spam call origins without generating CDR related traffic to the central CDR server. For our future works, we plan to investigate more spam call scenarios including the use of MVNOs. Another direction for further investigation is the application of AI agents to support the automation of call traceback processes and deal with difficult conditions.

REFERENCES

[1] H. Tu, A. Doupe, Z. Zhao, and G. Ahn, "Users Really Do Answer Telephone Scams", *Proc. 28th USENIX Security Symposium*, 2019.
[2] "AmericaUnder Attack:The Shifting Landscape of Spam and Scam Calls in America", True caller Insights, 2024.
[3] J. Mceachern and E. Burger, "How to Shut Down Robocallers: The STIR/SHAKEN protocol will stop scammers from exploiting a caller ID loophole", *IEEE Spectrum*, pp.46-52, Dec. 2019.
[4] J. Yu, "An Analysis of Applying STIR/SHAKEN to Prevent Robocalls", Proc. Advances in Security, Networks and Internet of Things, 2021.
[5] "Defeat Fraud Through Validation", AB Handshake Global Solution for Call Validation.
[6] A. Sheoran, S. Fahmy, C. Peng, and N. Modi, "NASCENT: Tackling Caller-ID Spoofing in 4G Networks via Efficient Network-Assisted Validation", *Proc. IEEE INFOCOM*, 2019.
[7] S. Pandit, K. Sarker, R. Perdisci, M. Ahamad, and D. Yang, "Combating Robocalls with Phone Virtual Assistant Mediated Interaction", *Proc. 32nd USENIX Security Symposium*, 2023.
[8] H. Deng, W. Wang, and C. Peng, "CEIVE: Combating Caller ID Spoofing on 4G Mobile Phones Via Callee-Only Inference and Verification" *Prof. ACM MobiCom'18*, 2018.
[9] S. Prasad, E. Bouma-Sims, A. Mylappan, and B. Reaves, "Who's Calling? Characterizing Robocalls through Audio and Metadata Analysis", *Proc. 29th USENIX Security Symposium*, 2020.
[10] L. Behan, J. Rozhon, J. Safarik, *et al.*, "Efficient detection of spam over internet telephony by machine learning algorithms", *IEEE Access*, December 2022.
[11] M. Azad, S. Bag, C. Perera, M. Barhamgi, and F. Hao, "Authentic-Caller: Self-enforcing Authentication in Next Generation Network", *IEEE Transactions on Industrial Informatics*, 2019.
[12] S. Wang, *et al.*, "Spoofing Against Spoofing: Towards Caller ID Verification In Heterogeneous Telecommunication Systems", *ACM Transactions on Privacy and Security*, 2023.
[13] D. Adei, V. Madathil, and S. Prasad, "Jager: Automated Telephone Call Traceback", *Proc. ACM SIGSAC Conference on Computer and Communications Security*, pp. 2042-2056. 2024.
[14] D. Geneiatakis, *et al.*, "Survey of security vulnerabilities in session initiation protocol", *IEEE Communications Surveys & Tutorials*, 2006.
[15] H. Lema, F. Simba, and J. Mushi, "Security Enhancement of SIP Protocol in VoIP Communication", *Journal of ICT Systems*, 2023.