Spam Call Detection with Hybrid Call ID Spoofing Detection and Cryptographic User Authentication

Jiawei Zheng, Jianhua He, Hsiao-Hwa Chen, *Life Fellow, IEEE*, Han Yang, and Tao Gao

***Abstract*—Caller ID spoofing (CIS) remains a major challenge in combating spam calls, as attackers can generate spam even with seemingly legitimate identifiers. Existing CIS detection (CISD) approaches, whether network-based or device-based, suffer a high false positive ratio and are limited in scalability and effectiveness. This paper proposes a comprehensive spam call detection framework that combines CISD with cryptographic user authentication (UA) to ensure secure verification of authorized callers. We introduce a hybrid CISD method, integrating network- and device-based approaches to exploit their complementarity, and a SMS-based CISD method, SMSpoof. We design a novel receiver-initiated UA scheme, ReCall, which leverages a UA server to validate caller identity and securely bind users to call IDs. By employing CISD and ReCall jointly, the framework achieves both higher detection accuracy and efficiency. Experimental results demonstrate the feasibility and effectiveness of the proposed approach, showing significant improvements in spam detection, communication overhead, and computational efficiency.**

***Index Terms*—Call ID Spoofing; Call Authentication; Spam Call; STIR/SHAKEN; User Authentication.**

I. INTRODUCTION

Compared to the legacy public switched telephone network (PSTN) technology, voice over IP (VoIP) offers substantial benefits, including flexibility, mobility, and cost savings, especially for long-distance and international communications. However, VoIP also introduces the problem of spam calls due to the lack of user authentication and ease of caller ID spoofing (CIS), which enables robocalls and scams. The low cost of VoIP calls makes it easier for spammers to make bulk calls inexpensively, while its global reach helps them to operate across borders without traditional barriers. Additionally, automated dialing systems (bots) can readily generate large volumes of spam calls. They have caused significant financial losses to individual victims and nationwide economy [1], and damaged the phone users' trust on the telephone systems and voice communications.

Jiawei Zheng (emal: 2238486002@qq.com) is with the School of Electronic Engineering, Beijing Post and Telecommunication University, China. Jianhua He (email: j.he@essex.ac.uk) is 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. Han Yang (email: h.yang29@lancaster.ac.uk) is with School of Computing and Communications, Lancaster University, UK. Tao Gao (email: gtnwpu@126.com) is with the School of Data Science and Artificial Intelligence, Chang'An University, China.

This work was funded by the European Union's Horizon 2020 research and innovation program under the Marie Skodowska-Curie Grant agreement Nos. 824019 and 101022280, Horizon Europe MSCA program under Grant agreement No. 101086228, and EPSRC with RC Grant reference EP/Y027787/1, and Taiwan National Science and Technology Council under Grant No. 114-2221-E-006-112.

Existing works tackling spam calls were focused mainly on call ID spoofing detection (CISD). CIS has been widely used to make spam calls, which deliberately changes the phone number and/or the name that is relayed as the caller ID information. An illustration of call spoofing is presented in Fig. 1. CIS 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. The existing CISD methods can be classified to end-devices based and network based methods, each targeting different layers of the telecom ecosystem [2]–[8].

Device based solutions do not rely on the network infrastructure and therefore are easy to implement and scalable. They can be further classified as callee only (such as blacklists using reported spam call numbers or machine learning based method to analyze the features of the calls) [1], [6], and end-to-end (E2E) methods, which requires an extra channel for interaction between the callees and the callers. Network based solutions depend only on the network infrastructure to play the roles of authentication and spoofing detection. They can be very effective and authoritative, making it easy to get industry and government support for a large scale deployment. STIR/SHAKEN (S/S) is one of the most representative and widely deployed network based solutions [2], which is mandated to be implemented by the service providers. However, the network based solutions require a significant investment for infrastructure upgrade and centralized management entities.

Despite the great efforts made so far, the existing CISD methods face an inherent limitation for spam call detection. Legitimate SIM cards can be obtained easily and spam calls can be made with legitimate IDs, which will be labeled by the CISD methods as legitimate calls. The trust on the CISD methods can lead to a worsen spam call impact. More secure call user authentication (UA) is needed for spam call detection, which has been rarely studied in the literatures. Unlike normal UA, call UA is more challenging as the calling parties may not know each other before the call. The callee may know only the call ID and even the call ID may be spoofed. Furthermore, the capability and capacity of telephone channels for the call UA are significantly constrained compared to normal UA.

In view of the aforementioned research challenges and gaps, in this work we propose a spam detection framework with CISD and call UA, based on which new CISD and call UA methods are designed. We first introduce a hybrid CISD approach, which integrates both network based and end-device based CISD methods, to exploit their complementarity

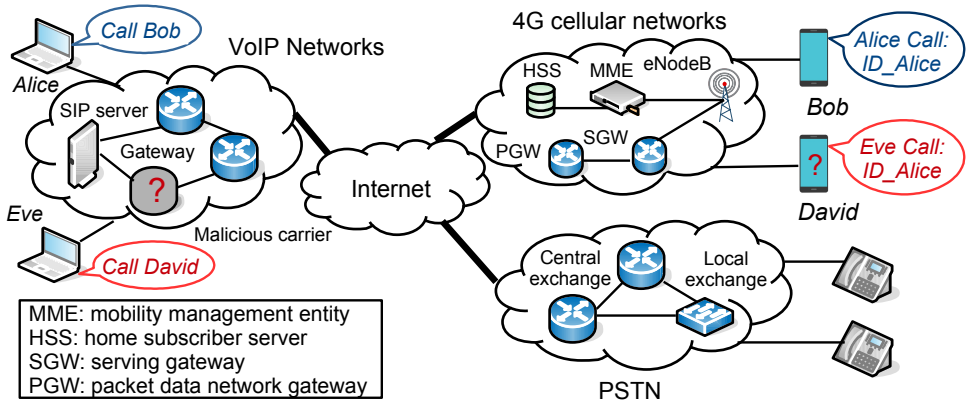

Fig. 1. An illustration of call spoofing, where Eve called David using VoIP devices from a VoIP network. This call ID is spoofed as Eve used the ID of a legitimate user Alice. The call might have gone through PSTN network before entering a 4G cellular network to which David is connected. It also shows that a mobile user Bob received a legitimate call from Alice with call ID displayed correctly.

for better scalability and CIS detection accuracy. Particularly, we propose a highly scalable short messages (SMS) based CISD method, i.e., called SMSpoof. Following the spoofing detection scheme of network based methods, the callee can use a SMS channel to challenge the caller for CISD. The caller is requested to respond to the callee with a response. As an example, we investigate the integration of S/S with SMSpoof to form a hybrid CISD method SSMSpoof, which achieves a significant performance improvement. On top of that, we propose a receiver-initiated call UA (ReCall) method, which leverages the cryptographic UA and the assistance of a central UA server. The callers register their certificates which include their call numbers to the UA server. Upon receiving a call, the callee can check if the caller supports UA and performs UA process via the UA server. The proposed UA method can work along with the CISD method to improve the spam call detection performance and system efficiency. Simulations were performed to evaluate the proposed works. Experiment results showed that the proposed framework is highly efficient and scalable. By integrating SSMSpoof with ReCall, a significant performance improvement is achieved over the existing UA method, in terms of spam call detection, communication, and computation efficiency.

The remainder of this paper can be outlined as follows. Section II reviews the related works. Section III introduces the proposed CISD methods along with the call UA methods. Section IV presents the performance evaluation, and Section V concludes the paper.

II. EXISTING CISD AND CALL USER AUTHENTICATION SCHEMES

In this section, we will give an overview of the existing CISD and call UA methods.

A. End-device based CIS Detection Methods

End-device based methods conduct spoofing detection by the callees with or without the involvement of the callers, where a telephone network can be considered as a black box. As they do not need the support from the network infrastructure, they are more scalable and can be deployed flexibly without a significant infrastructure investment.

1) Callee Only Methods: Caller reputation and blacklist based methods have been widely used in practice for CISD. A major limitation of these methods is that they are effective with reported malicious phone numbers but not with new numbers. The spammers can also check the blacklists and spoof with other non-reported new numbers or use new phone numbers which can obtained easily.

CEIVE is a callee only inference and verification method for combating CIS [6]. It leverages a callback session and its associated call signaling to infer the call state of the caller. By comparing the anticipated call state, it verifies whether an incoming call comes from the originating number. It is simple and shows a high accuracy in the experiments. However, it could be difficult to access the call signaling messages. Furthermore, as it uses a trained classifier to infer the call states, which is affected by the phone carriers, and could be a big problem for phone calls from new carriers.

A machine learning algorithm based method was proposed in [9], using call data records collected from mobile phones and public datasets to classify spam calls. This method is simple, but its classification accuracy and generalization ability are limited. The call data records may not present sufficient information for accurate detection of CIS and spam calls.

2) E2E Methods: CallerDec is one of the first E2E methods to detect CIS with its two variants, i.e., timing-CallerDec and SMS-CallDec, which use automated callback and SMS, respectively [10]. Timing-CallerDec creates a trusted covert channel for call signaling. After receiving an original call, the callee will initiate a new verification call by calling the caller and the caller is required to respond to the verification call. Timing-CallerDec is simple but sensitive to a predefined parameter of time waiting to reject a verification call, and therefore, it is not robust and has limited practical use. In the SMS-CallerDec, the caller sends SMS to the callee with call intent before making a voice call. The callee sends an SMS to challenge the caller upon receiving the SMS from the caller. This method requires the support from both caller and callee, which may limit its usability. And the caller needs to send an

intent SMS first, which may not happen for spoofed calls.

Spoof Against Spoofing (SAS) method was proposed with improved robustness by using a challenge and response mechanism [8]. The callee uses a spoofing to make a verification call with a challenge, requesting the phone user with the calling number to call back and respond with the shared secret. While the SAS approach was demonstrated to work with devices connected to different types of phone networks, it requires the application of dual-tune multi-frequency (DMTF) technique by the caller to convey communication information, which may not be easy to implement. Furthermore, with an increasingly wide deployment of S/S, the verification calls with spoofing may be attested with a low trust level and thus blocked.

B. Network Infrastructure Based CIS Detection Methods

S/S method is a network based CISD method [2], which was designed to manage the deployment of secure telephone identity technologies to provide E2E cryptographic authentication and verification of telephone identity and other information in VoIP networks. It provides a framework for telephone service providers to create and verify signatures in SIP. The originating carrier provides an attestation of the caller numbers with one of three possible levels (i.e., level A for full attestation, B for partial attestation, and C for gateway attestation). These three levels reflect a decreasing confidence of the carrier on the caller number. The signing service provider signs the attestation and embeds its signature in the SIP INVITE signaling message. If the carriers in the call path from the originating carrier to the terminating carrier all support the STIR/SHAKEN protocol, the attestation level will be forwarded to the callee, which will help the callee to decide whether accepting the call or not.

AB handshake protocol is an alternative network based solution. Instead of requiring the support from all the carriers in the S/S protocol, AB handshake is more scalable and flexible than S/S, requiring only originating and terminating carriers to verify the legitimacy of the callers using secure API exchanges with the help of bilateral cooperation between the carriers [3]. But it faces a major challenge of discovering the originating carrier by the terminating one. In addition, while it is easy to start the operation, providing full CIS detection services from all the pairs of carriers could be more complex and costive.

On the other hand, NASCENT tackles CIS in 4G networks with network assisted validation [11]. It argues that even when mobile user authentication is in place, CIS is feasible during the call setup (e.g., different IDs may be used in the SIP INVITE message). It leverages the subscriber data already available to the evolved packet core (EPC) and cross validates the caller ID of an incoming voice call at the IP multimedia subsystem (IMS). One limitation of NASCENT is that both the callers and the callees need to be mobile users. And the IMS where the call ends and the EPC where the call originates need to be from the same service provider; otherwise, it is difficult for the IMS to access the subscriber data at the EPC.

C. Spam Call Detection and Telephone User Authentication

AuthLoop uses an E2E cryptographic authentication for telephony over voice channel [12], where the caller and callee use in-band modem to execute a transport level security (TLS) inspired authentication protocol. While it could provide strict user authentication, the available bandwidth and reliability of the voice channel are highly limited. In addition, the implementation of the protocol is specific to telephone devices, which can present a big challenge.

AuthCall was built upon AuthLoop by utilizing a separate data channel for fast cryptographic authentication [13]. To initiate a call, the caller sends a message to a central server, indicating the intent to contact the callee. The server then facilitates the authentication process between the caller and the callee. Since the intent message must be sent to the server and the server is required to coordinate the authentication process for every call, the resulting traffic and computational load can be overwhelming, potentially making the server a performance bottleneck.

UCBlocker is an E2E solution designed to block unwanted calls [14]. It relies on a new infrastructure based on anonymous credentials, which enables anonymous caller authentication and policy definition. The users are allowed to define the policies of acceptable calls. This design decouples caller authentication and call session initiation. A verification code from the callee is used to interface and bind the two processes. A potential problem is on the design of polices to block all unwanted spam calls out of the callee's preference without missing important legitimate calls.

Recently, large language models (LLMs) were explored for real-time detection of phone scams [15], in which scam calls are modeled and LLM is applied to assess fraudulent intent in telephone conversations. Immediate warnings can be provided to users to mitigate the harm of scam calls. As the LLMs are used to monitor and analyze the call conversations, the traffic overhead and cost could be very high, especially for long legitimate calls. In addition, the LLMs may not be adaptive to the new phone scams with updated patterns.

III. PROPOSED SPAM CALL DETECTION FRAMEWORK

The network based methods especially S/S are the major and promising CISD. However, the existence of legacy carriers which do not support S/S can largely affect the effectiveness of S/S method. Furthermore, CISD alone cannot solve the spam call problems completely as spammers can use legitimate call numbers to make spam calls without being detected. On the other hand, the existing user authentication methods for spam call detection are not efficient and scalable. New and more efficient spam call detection methods are needed. In this section, we present a new framework for spam call detection with improved CISD and novel call UA methods.

A. A New Framework of Spam Call Detection

In the proposed new framework, both CISD and UA are included with its design rationale been explained as follows. As CISD alone cannot completely prevent spam

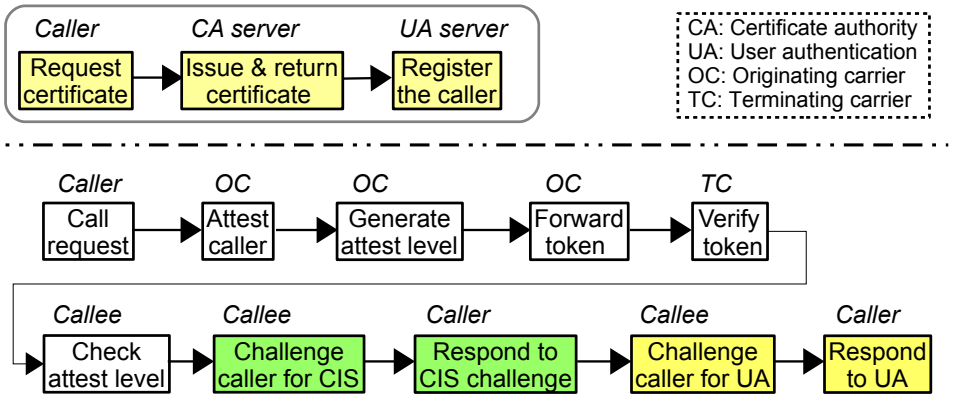

Fig. 2. A new framework of spam call detection with hybrid CISD and cryptographic UA. Before making calls, callers undergo the certification application process illustrated at the top of the figure. During call setup, the calls are subject to CIS detection and caller authentication..

calls due to the possible use of legitimate numbers, UA is necessary for reliable and complete spam call detection. While cryptographic UA can reliably detect spam call, using UA alone for spam call detection can be unnecessarily inefficient and non-scalable. First, without CISD, telephone users need to implement UA methods for spam call detection. It can be difficult to motivate them to do so. Second, for the spam calls using CIS, they could be detected effectively and reliably by the CIS detection methods, and therefore, no further UA is needed, which can save significantly communication and computational resources. In addition, not all of the calls with unknown call numbers or names are important to the callees. They could accept the calls attested by the S/S method with level A and reject the calls attested with level B without going through the cryptographic UA process, which will also reduce traffic and computational loads at the central authentication servers.

The overall process in the proposed framework is introduced with an illustration as shown in Fig. 2. It is noted that the blocks with orange background represent the operations related to UA. The blocks with white background represent the operations related to network based CISD, and those with green background represent the operations related to device based CISD. For the UA task, the callers who want their calls to be authenticated will first request certificates from a certification authority (CA) server. The certificates will include the telephone number, public key of the callers, and other relevant information such as name, organization and address. The certificates are signed by the CA server and sent to the callers. After the callers receive their certificates, they can register to a central UA server, which will store the certificates and support the late UA process. The certificate creation and registration processes are one-off process.

To make a call (either normal or spoofed), the caller should first request a call to the callee, which then goes through the optional CISD (with network based and/or end-device based). As shown in Fig. 2, the originate carrier (OC) attests the call and generates attestation level. The attestation level is added to a token, which is included in the SIP message header. The terminating carrier (TC) verifies the token and forwards the caller ID and attestation level to the callee. Based on the received call information, the callee decides whether to challenge the caller for CIS using optional E2E method or not. If yes, the caller needs to respond to the CIS challenge. The callee will check the refined CISD outcome and decide if further UA is needed. If yes, the callee will challenge the caller for UA and the caller will respond to the challenge. The callee will then have a clear view on the call and make a more well-informed decision to accept or reject the call. Note that it is possible that the caller and/or the callee do not support the device based CISD or the UA method, and then the call setup process falls back to the traditional one.

B. Proposed Device Based CISD Method: SMSpoof

In this subsection, we present a simple SMS based CISD method, which is called SMSpoof. As the network based CISD methods face the issues of partial and gradual implementation by the service providers and the existence of non-compatible legacy service providers, the SMSpoof is proposed to complement the network based CISD methods. For the spam calls, while the caller IDs may be mobile numbers, landline telephone numbers or VoIP numbers, most of the telephone numbers of the callees of spam calls are mobile numbers. On receiving a call with an unknown call number, if the attestation level is lower than A in case the network based CISD is in place, or the attestation level is not available, the callee can initiate an SMS based CISD, by sending a SMS with a randomly selected challenge, such as a number of 1324, to the caller. A timer is set to wait for the response from the caller. If the caller number is not spoofed, the caller should return an SMS to the callee with a response of the same number as the challenge. If the callee receives the response before the timer times out, the response will be compared to the challenge. If they are the same, then no CIS can be assumed; otherwise, CIS is detected. If no response is received from the caller before the timer times out, the callee can request a response from the caller over the telephone session, either automatically or manually. One advantage of SMSpoof is that the callers may not need to implement any app program for SMSpoof, just needing to follow the instruction of returning the received challenge. SMSpoof is simple but also faces some limitation. For instance, the caller

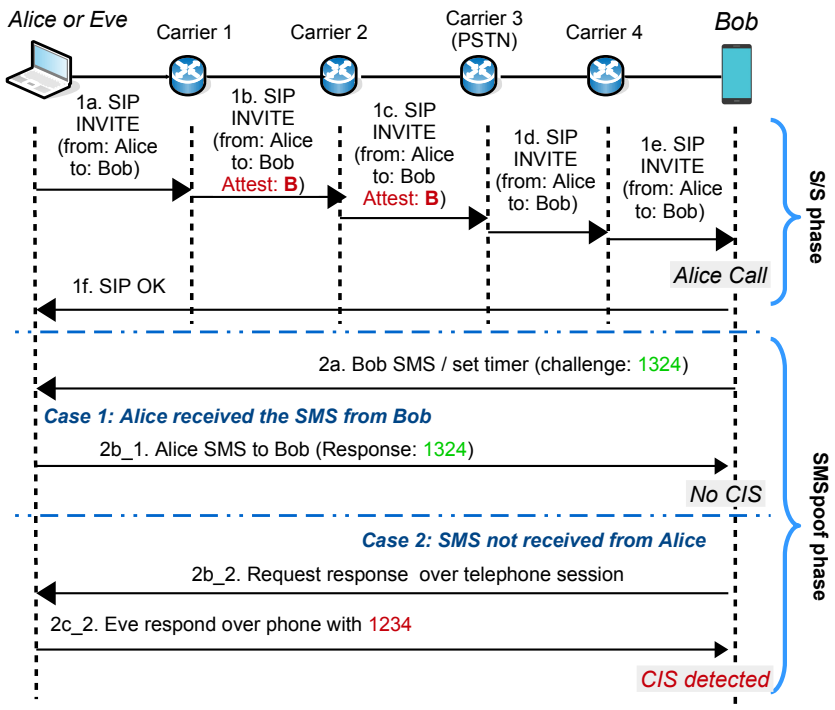

Fig. 3. Time flow of end-device based CISD method SMSpoof. During the S/S phase, a caller (either the legitimate Alice or an adversary Eve) places a call to Bob using Alice's call ID. Bob receives the SIP INVITE but sees no attestation information. In the SMSpoof phase, Bob issues an SMS challenge containing a one-time code (e.g., 1324) to the phone number associated with the presented caller ID. Two possible outcomes are presented. In Case 1 Alice receives the SMS and returns the correct code 1324. In Case 2 Eve is the caller who does not control Alice's mobile device and responds with an incorrect code (e.g., 1234). The mismatch triggers a CIS detection.

needs to be capable of receiving SMSs. In addition, relatively long delay of delivering SMSs makes it difficult to ensure completion of the SMS based CIS detection before the callee answers the call. SMS delivery delays often range from a few seconds to several minutes. In cases where calls originated from non-mobile devices or the call ends before the SMS messages are delivered, the SMSpoof method falls back to the traditional call process, leaving spoofing unmitigated. One possible solution to address this limitation is to require callers to register a mobile phone number with a central server. Callees could then query this server to verify whether a caller's mobile number is registered and, if desired, issue a verification challenge through that channel.

It is noted that SMSpoof method can work alone or with the network based methods. The method can be used automatically or manually. If the SMSpoof method works with network based CISD methods, we call the integrated method as a hybrid method. In this paper, we consider a hybrid CISD method running on S/S and SMSpoof, which is called SSMSpoof. A space-time diagram of the signaling flow for the SSMSpoof method is shown in Fig. 3. In the diagram, Alice requests a call to Bob, which goes through the S/S CISD process. As Carrier 3 does not support the S/S protocol, the attestation level information is removed from the signaling message. Eventually, the callee Bob does not receive the attestation level information. Bob can choose to use SMS for further CISD. It sends a SMS to Alice and detects CIS according to the response from Alice, with two cases of receiving the response or not, as shown in the diagram. SSMSpoof is the first such effort of integrating the network and device based CISD. Other end-device based methods could be integrated with network based ones.

C. Proposed User Authentication Method ReCall

Cryptographic UA can provide a strict and secure approach for spam call detection. Specific callers are first certified by a CA server, who are bound to the telephone number that is used to make a call. We argue that these callers are self-motivated to be certified; otherwise, their calls may be treated as spam calls that are more likely to be ignored by the callees. In this paper, we propose an efficient and effective cryptographic UA method, which is called ReCall. In the UA system, there are four major components, i.e., callers, callees, a CA server, and a UA server. The CA server is responsible to handle certificate requests and grant certificates to the users who meet the requirements. The UA server will actively participate in the UA authentication process. The CA server and UA server could be located in the same place. It is noted that UA is a fundamental building block for computer and network security. In the traditional UA systems, a user presents an identification to the security system for verification. In the proposed receiver-initiated call UA method, there is no such security system for verification. The traditional UA can not be applied directly.

Next, we will present the overall signal flow of the ReCall method. A space-time diagram of the signal flow is shown in Fig. 4, in which there are three phases, i.e., certificate creation, call setup, and UA phases, respectively. In the certificate creation phase (corresponding to Steps 1a to 1c), the caller

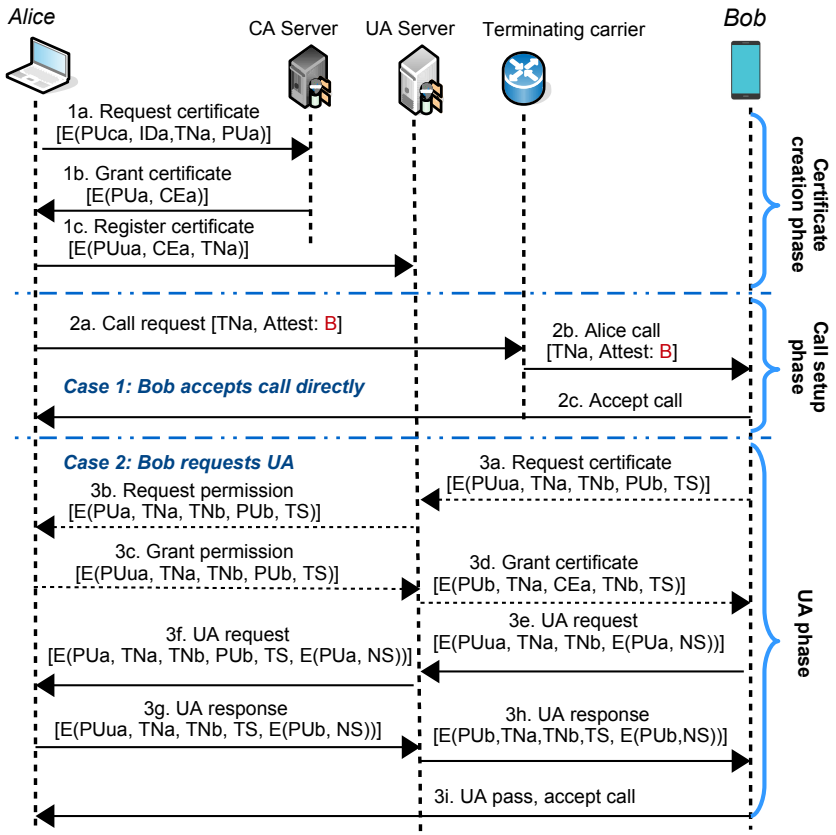

Fig. 4. Space-time diagram of signaling flow for the proposed UA method with certificate creation, call setup and UA phases, where E(PUx, y,z) denotes encryption of message with content of y and z by public key PUx. TNx denotes telephone number of user x.

(Alice in this example) first requests a certificate (or more if needed) from the CA server. It presents to the CA server its ID (IDa in the example), its telephone number (TNa), and its public key (PUa), which are encrypted by the CA server's public key. Therefore, only the CA server can decrypt the encrypted information. The CA server will grant a certificate (CEa) to Alice, which is sent to Alice encrypted by Alice's public key. Alice then registers to the UA server with the certificate CEa, which is encrypted by the UA server's public key PUua.

In the call setup phase (i.e., Steps 2a to 2c), the call request will go through the normal procedure, including optional S/S attestation and device based CIS detection. On receiving a call request, the attestation level information may be presented or not. If the callee supports the UA, he/she will decide whether to perform UA or not. In this example, we assume that the call ID is attested by the originating carrier as level B, which is forwarded to Bob. If Bob trusts the call, he can accept the call and end the call setup phase (i.e., Case 1 shown in the diagram). If Bob does not support the proposed UA method, he may reject the call. If Bob supports the proposed UA method, he can challenge the caller for the UA purpose, which moves to the UA phase.

In the UA phase (i.e., Steps 3a to 3i), one problem is that Bob may not know if the caller supports the proposed RUA methods. To solve the problem, we use a public UA server, which stores the certificates with the telephone numbers of the registered users. The callee Bob can download certificates of registered users from the UA server in advance. If the call number is shown in any of the downloaded certificates, Bob can send a message to the UA server to request the certificate associated with the received telephone number TNa. Bob's number TNb, public key PUb, and the timestamp (TS) are also included in the message, which are encrypted with the UA server's public key PUua for message confidentiality. If the certificate of Alice is available at the UA server, the UA server may share it with Bob directly. Alternatively, as shown in the example, the UA server optionally requests a permission from Alice for sharing certificate with Bob (Step 3c), with the information TNa, TNb, PUb, and TS included in the request. If Alice initiated the call, she could grant the permission; otherwise, she may decline the permission. With the permission from Alice, the UA server shares Alice's certificate with Bob (Step 3d), which is encrypted by Bob's public key PUb. Then, Bob sends a UA request to the UA server, which is forwarded to Alice. In the request, a nonce NS is encrypted with Alice's public key PUa, which only ensures Alice can decrypt the cyphertext to access the NS. After receiving the UA request, Alice will respond to the request with a message including the NS, which is encrypted by Bob's public key PUb (Step 3g). The response is sent to the UA server, which is forwarded to Bob (Step 3h). Bob verifies the response by comparing the sent NS and received NS. If they are equal, the verification process is passed and Bob can trust and accept the call (Step 3i).

It is noted that for a caller who has been authenticated by a callee, the caller's certificate can be cached. If the callee receives a new call from the caller in the future, the cached certificate can be reused without being requested from the UA Server. Furthermore, if the new call passes the CISD with an attestation level C, the callee can ignore the call and there is no need to request UA, which can reduce communication and computation loads.

IV. SECURITY AND EFFICIENCY ANALYSIS

In this section, we analyze and evaluate the performance of the proposed CISD and UA methods on spam call detection. First, we compare the security performance and the communication traffic loads of the ReCall and the AuthCall authentication methods. Then, spam call detection performances with CISD methods will be compared in terms of true positive rate (TPR), true negative rate (TNR), and false positive rate (FPR). They are denoted by P_{tp}, P_{tn}, and P_{fp}, respectively, and all are in the range of 0 to 1. Last, the UA methods are also compared in terms of P_{tp}, P_{tn}, and computational loads on UA.

A. Security Analysis of ReCall and AuthCall UA Methods

ReCall is built on the cryptographic UA protocol with the help of a CA server and an optional UA server. It can achieve fully secure UA performance. In the key steps of ReCall protocol (i.e., Steps 3e to 3i of Fig. 4), the UA request and response messages are transmitted via the UA server if a direct communication channel between the caller and the callee is not available. As the messages are encrypted using the public keys of the UA server and the calling parties, the confidentiality of these messages is preserved. Furthermore, as the challenge number NS in the request message used for UA is encrypted by the caller's public key, only the caller can view the number NS. In the response message, as the NS is encrypted by the callee's public key, only the callee can view the number NS returned by the caller. Therefore, the security properties of confidentiality, integrity and digital signature for the call UA process are preserved. One potential risk arises if an adversary gains access to both the mobile device and the private key of a registered legitimate caller. In this case, the adversary could generate spam calls that appear authentic and may be accepted by callees. Nevertheless, such an event is considered highly improbable, and any resulting spam calls can be reported, enabling the adversary to be traced and identified.

AuthCall is the only known existing call UA method, which uses a similar system structure [13]. It can also achieve a fully secure UA performance. The main difference between AuthCall and the proposed ReCall lies on the fact that the AuthCall is caller initiated, while ReCall is callee initiated. Using AuthCall, the caller makes UA request for every call via the UA server, no matter if the callee is available, supports UA, or is willing to accept the call and the UA request. The UA server will need to process many unnecessary UA related messages, which can become a bottleneck of the UA system. And usually, it is the callees who are more motivated to verify calls using UA rather than the callers.

ReCall is much more efficient than the AuthCall, which can be illustrated with the following simple example. Assume that a callee may support call UA with a probability denoted by P_{ua}. A callee that supports call UA may choose to perform UA for an incoming call with probability P_{w}. Let $N_{\text{m, authcall}}$ and $N_{\text{m, recall}}$ denote the average numbers of messages received or transmitted by the UA server for AuthCall and ReCall, respectively. Following the signal flow as shown in Fig.4 of [13], if the callee does not support call UA, the UA server will process two messages. If the callee supports UA but declines the UA request, the UA server will process three or four messages. If the callee accept the UA request, the UA server will process seven messages. Then, $N_{\text{m, authcall}}$ can be computed by $N_{\text{m, authcall}} = 2(1-P_{\text{ua}})+3P_{\text{ua}}(1-P_{\text{w}})+7P_{\text{ua}}P_{\text{w}}$.

Because the certificates of the callers supporting UA can be downloaded by the callees in advance and an optional permission request of the caller certificate is not needed, there are only four messages to be processed by the UA server. Therefore, $N_{\text{m, recall}}$ can be computed by $N_{\text{m, recall}} = 4P_{\text{ua}}P_{\text{w}}$. Assuming $P_{\text{ua}} = 0.3$ and $P_{\text{w}} = 0.4$, we have $N_{\text{m, authcall}} = 2.78$, and $N_{\text{m, recall}} = 0.48$. The UA server will need to process more than 4.8 times messages than AuthCall.

Using central server in the call UA methods could affect the capacity and the call response performance of the system, under the constraints of communication and computing resources in the server. Given the communication and computation loads of the call UA methods, the system capacity of handling call UA requests can be computed. Consider a real telephone system with 10,000 calls per second and a communication link of 800 Mbps for a central UA server [4]. According to the analysis that on average the UA server will receive 0.48 UA messages for a call. Suppose a UA message has a length of 1000 bytes. Then, the server communication traffic is computed by $10^4 \times 10^3 \times 8 \times 0.48 \times 10^{-6}$ = 38.4 Mbps, which is far below the UA communication link capacity.

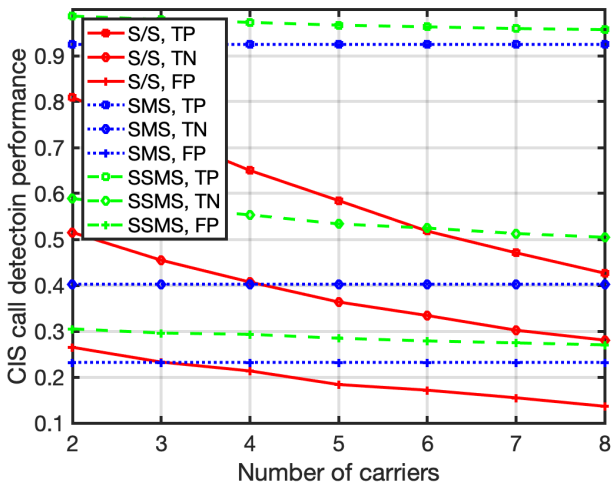

Fig. 5. Spam call detection results with three CISD methods against the number of carriers in call path, in terms of TPR, TNR and FPR. SMSpoof method is shown as SMS in the figure.

B. Performance Evaluation of CISD Methods

CISD methods can be used to detect spam calls, although the spam calls using legitimate numbers will not be detected. Next, we compare S/S, SMSpoof and hybrid method SSMSpoof by simulations. The number of carriers in the call paths (including the originating and terminating carriers) varies from 2 to 8. These carriers are classified to S/S carriers (supporting the S/S protocol) and legacy carriers (not supporting S/S). The probability that a carrier is legacy carrier (denoted by P_{leg}) is set to 0.1. The probability that a spam call uses spoofing is 0.6. For the SMS based CISD method, the probability that both the caller and the callee of a call support SMSpoof is 0.8. Every mean value was obtained via 10,000 simulations.

The spam detection results of the three compared CISD methods are presented in Fig. 5. It can be observed that the SSMSpoof performs the best in terms of TPR and TNR. SMSpoof performance is worse but close to that of SSMSpoof. As SMSpoof is an end-device based scheme, its performance is not affected by the number of carriers. The performance of S/S method degrades with an increasing number of carriers. It is also observed that all the three CISD methods have high FPR, which can have a large negative impact on the callees as they may be misled by the assurance of the call trust by the CISD results. And SSMSpoof has the highest FPR, with $P_{\text{fp}} = 0.3$ under the setting of two carriers. The high FPR observed in the three compared CISD methods is due primarily to adversaries exploiting legitimate numbers to initiate spam calls. The above results clearly show the limitations of the CISD methods and the need of more effective spam call detection methods.

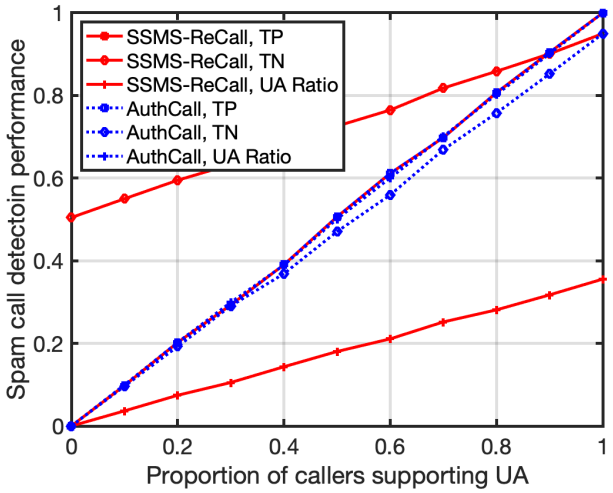

Fig. 6. Spam call detection results with UA methods, ReCall and AuthCall, in terms of TPR, TNR and the ratio of calls performing UA.

C. Spam Call Detection Performance with UA Methods

Next, we compare the AuthCall and ReCall methods on spam call detection in terms of TPR, TNR and the ratio of calls that perform UA process (UA Ratio). The UA Ratio metric is used to measure the computation and communication loads at the caller, callee, and the UA server. The SSMSpoof CISD method is integrated into ReCall for spam detection, and the integrated method is called SSMS-ReCall. The operation of the method is presented below. If CIS of a call is detected by SSMSpoof, the call is classified as a spam call and the ReCall UA is not needed. If no CIS is detected by SSMSpoof and the call ID is in the contact list of the callee, then the call is classified as trusted and ReCall UA is not needed. Otherwise, if the callee supports ReCall and decides to run a call UA, it requests a UA with the caller via the UA server. If the UA is passed, the call is verified to be a legitimate call; otherwise, it is verified to be a spam call.

For the system level simulations, the settings for the SSMSpoof method are the same as those in the previous subsection. Assume that the callee uses a mobile phone number and the probability that the caller uses a mobile number with SMS capability is 0.8. The probability that a call is spam is 0.4 and 0.8 for the cases that the caller uses a mobile number and a non-mobile number, respectively. The above setting reflects the fact that more spam calls use landline numbers. For a call with a mobile number, the probability that the call number is in the callee's contact list is 0.8 and 0.1 for the cases that the call is a legitimate and spam call, respectively. For a call with non-mobile number, the probability that the call number is in the callee's contact list is 0.6 and 0.3 for the cases that the call is a legitimate and spam call, respectively. It is noted that we used some fixed parameters that are configurable with different values. While real-world datasets could better demonstrate robustness, only small datasets are available, and most focus on spam call detection using call contents. Such datasets are not suitable for configuring or conducting our simulations.

The results of the SSMS-ReCall and AuthCall methods are presented in Fig. 6. It can be observed that SSMS-ReCall and AuthCall methods have the same TPR performance, which are linear against the proportion of the callers supporting UA (denoted by P_{cua}). However, SSMS-ReCall performs significantly better than AuthCall in term of TN performance. For example, for $P_{\text{cua}} = 0.6$, the TNR is 0.8 and 0.6 for ReCall and AuthCall, respectively. Note that as FPR is zero for both UA methods, it is not presented in Fig. 6. SSMS-ReCall also enjoys a much lower computational load (for processing UA related messages) than AuthCall. For instance, for $P_{\text{cua}} = 0.6$, the ratio of calls performing UA is 0.2 for SSMS-ReCall, which is one third of that (i.e., 0.6) for AuthCall. The above results demonstrated a superior performance of SSMS-ReCall against AuthCall and the effectiveness of integrating CISD method with cryptographic UA methods for spam call detection.

V. CONCLUSION

Spam calls cause significant economic and social damages. CISD is the most well-known scheme to combat spam calls. However, it is not very effective and has its inherent limitations. In this paper, we proposed a new spam call detection framework, which combines both CISD and cryptographic UA to exploit their complementarity for better detection accuracy and efficiency. To improve the CISD

performance, a hybrid CISD method was designed, which integrates both network infrastructure based and end-device based methods. As the CISD methods suffer a high false positive rate, a new receiver initiated call UA method (i.e., ReCall) was proposed with the CISD methods integrated. Experiment results demonstrated feasibility and effectiveness of the proposed spam call detection framework. In addition, ReCall is more effective and efficient than the existing caller initiated UA method AuthCall in terms of spam call detection, communication and computation loads. Future work includes further investigating and enhancing the scalability of the call authentication scheme, potentially through distributed authentication systems. In addition, user studies or surveys will be conducted to systematically assess the impact of the proposed methods on user experience, complementing the technical performance evaluation. Another important direction is developing automated approaches to trace the origins of spam calls.

REFERENCES

[1] "AmericaUnder Attack:The Shifting Landscape of Spamand Scam Calls in America", Truecaller Insights, 2024.

[2] J. Mceachern, 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.

[3] "Defeat Fraud Through Validation", AB Handshake Global Solution for Call Validation.

[4] 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.

[5] J. He, H. Chen, K. Yang, T. Gao, Z. Cao, "Automated Spam Call Traceback: Two Efficiency Enhancement Approaches". IEEE Network. July 2025.

[6] H. Deng, W. Wang, C. Peng, "CEIVE: Combating Caller ID Spoofing on 4G Mobile Phones Via Callee-Only Inference and Verification," *Prof. ACM MobiCom'18*, 2018.

[7] M. Azad, S. Bag, C. Perera, M. Barhamgi, F. Hao, "Authentic-Caller: Self-enforcing Authentication in Next Generation Network", *IEEE Transactions on Industrial Informatics*, 2019.

[8] S. Wang, *et al.*, "Spoofing Against Spoofing: Towards Caller ID Verification In Heterogeneous Telecommunication Systems", *ACM Transactions on Privacy and Security*, 2023.

[9] L. Behan, J. Rozhon, J. Safarik, F. Rezac, M. Voznak, "Efficient Detection of Spam Over Internet Telephony by Machine Learning Algorithms", *IEEE Access*, Dec. 2022.

[10] H. Mustaf, W. Xu, A. Sadeghi, and S. Schulz, "End-to-End Detection of Caller ID Spoofing Attacks", *IEEE Transctions on Dependable and Secure Computing*, Vol. 15, No. 3, May-June 2018.

[11] A. Sheoran, S. Fahmy, C. Peng, N. Modi, "NASCENT: Tackling Caller-ID Spoofing in 4G Networksvia Efficient Network-Assisted Validation", *Proc. IEEE INFOCOM*, 2019.

[12] B. Reaves, L. Blue, P. Traynor, "AuthLoop: End-to-End Cryptographic Authentication for Telephony over Voice Channels", *Proc. usenix*, August 2016.

[13] B. Reaves, *et al.*, "AuthentiCall: Efficient Identity and Content Authentication for Phone Calls", *Proc. usenix*, August 2017.

[14] C. Du, Y. Xiao, Y. Hou, A. Keromytis, W. Lou, "UCBlocker: Unwanted Call Blocking Using Anonymous Authentication", *Proc. usenix*, August 2023.

[15] Z. Shen, *et al.*, "It Warned Me Just at the Right Moment: Exploring LLM-based Real-time Detection of Phone Scams", *arXiv:2502.0394v1*, February 2025.