IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 20, 2017, accepted October 13, 2017, date of publication October 30, 2017,
date of current version November 28, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2764955

TrustCall: A Trust Computation Model for Web Conversational Services

IBRAHIM TARIQ JAVED, KHALIFA TOUMI, AND NOEL CRESPI
Institut Mines-Telecom, Telecom SudParis, 91000 Evry, France
Corresponding author: Ibrahim Tariq Javed (ibrahim_tariq.javed@telecom-sudparis.eu)

ABSTRACT Web conversational services are exposed to several threats in which the social context between communicating participants is manipulated. Cybercrimes based on identity misrepresentation to obtain sensitive information are on the rise. Various scams and frauds are conducted by distributing malicious content, viruses, and spam over established communication sessions. In order to maintain overall security and enhance privacy, methods of estimating trustworthiness and reputation should be built into Web calling services. In this paper, we propose ''TrustCall'' a reputation-based trust model for real-time Web conversational services. In our approach, the reputation of a caller is evaluated using *Authenticity Trust* and *Behavioral Trust*. *Authenticity Trust* describes the legitimacy of a caller by collecting recommendations from other members of the network, whereas *Behavioral Trust* determines a caller's popularity based on its communication behavior. Other contributions include a threat taxonomy for Web calling services, including social threats, that directly target users. A set of experiments are conducted in order to prove the feasibility and effectiveness of our model.

INDEX TERMS WebRTC, trust, reputation, recommendation, popularity, trust model, social trust.

I. INTRODUCTION

Web conversational services allow users to have real-time audio and/or video calls and direct data transfers. Over-The-Top (OTT) operators such as Google, Skype and WhatsApp offer cost-effective and innovative services compared to the traditional calling services provided by telecom operators. With the introduction of the WebRTC (''Web Real-Time Communication'') standard [1], any website can now offer communication services in a ubiquitous manner. WebRTC is an aggregation of protocols and Application Programming Interfaces (APIs) that enable real-time communication between users in a Peer-to-Peer (P2P) fashion. WebRTC is expected to boost Voice-Over-IP (VoIP) into novel decentralized communication platforms offering cross-domain interoperability and identity portability [2], [3]. Telecom operators intend to adopt WebRTC technology in order to compete with existing OTT web conversational services [4]–[6]. WebRTC is thus considered by many to be a revolutionary market disruption for telecom industry.

However, web-based conversational services are exposed to several threats in which the social context between communicating participants is manipulated. In traditional telecommunication networks, subscribers are identified by the use of geographically recognizable phone numbers, but web-based identities are simply a combination of self-created user profiles and credentials. Attackers generally misrepresent themselves by presenting fraudulent information over calls in order to conduct numerous scams and frauds [7]. Furthermore web communication services are exposed to threats in which viruses, spywares, illegal content and spam are distributed. Web-based Communication Service Provider (CSP) must provide solutions to prevent the misuse of conversational services. Methods of estimating trustworthiness and reputation should therefore be built into future WebRTC-based calling services.

Several researchers have presented security and trust requirements for WebRTC standard [8]–[10]. It is desirable for a user to be certain to whom they are speaking. Therefore, WebRTC identity architecture allows communicating participants to identify and authenticate each other before establishing a communication session [8]. Authors in [9] discuss different models for provisioning user identity and their impact on user privacy. Meanwhile, the authors in [10] provide mitigating techniques and security mechanisms against identity fraud conducted while establishing a communication session. Identification is the first step in recognizing a user over a communication session. However, it cannot guarantee the trustworthiness of a user. User authentication alone is not

enough to secure communication services from the various social security threats that are present over communication networks. Therefore, new mechanisms must be introduced to ensure that callers over web communication networks act in an acceptable, responsible and legitimate manner.

We present a new reputation-based trust model to estimate the trustworthiness of communicating participants. The computed trust is used to differentiate between legitimate and malicious callers over web communication services. *TrustCall* is a hybrid trust computational model based on the evaluation of *Authenticity Trust* and *Behavioral Trust*. *Authenticity Trust* describes the legitimacy and genuineness of a caller's identity whereas *Behavioral Trust* determines the popularity and acceptance of a user in the network. *Authenticity Trust* is based on recommendations received by other members of the network, while *Behavioral Trust* is computed by examining the communication behavior of the user. The feasibility and effectiveness of the model is shown using a simulated network of communicating peers. A detailed description of the potential social threats that exist in real-time web communication services is also presented.

The rest of the paper is structured as follows: the related work is described in Section II. A threat taxonomy for real-time web conversational services is presented in Section III. The identity architecture of WebRTC standard is examined in Section IV. The three components of '*TrustCall*' model: information collection, trust computation and trust usage are described in Section V. The evaluation of *Authenticity Trust* is provided in Section VI, whereas the *Behavioral Trust* is explained in Section VII. In Section VIII, various experiments are conducted to prove the feasibility and effectiveness of the TrustCall model. Finally, our conclusions and recommendations for future work are provided in Section IX.

II. RELATED WORK

This section provides a comprehensive literature review of the WebRTC standard, as well as an analysis of the existing trust computational models.

WebRTC is a set of protocols that enable real-time communication between communicating participants [1]. The identity specifications of WebRTC are highly flexible when compared to the closed ecosystems of existing VOIP solutions. The WebRTC architecture [8] allows communicating participants to identify each other before establishing a communication session [11]. Each user verifies the authenticity of a communicating participant's identity independent of the service provider [12]. Different models for provisioning user identity in an end-to-end manner are defined in [9]. Authors in [10] have proposed several mitigating techniques and security improvements for WebRTC identity specifications, and new requirements for WebRTC identity architecture are highlighted in [13] and [14].

A considerable amount of literature has been published on identifying and authenticating users over WebRTC-enabled services. For instance, the researchers in [15] provide authorization models based on access control lists and capability-based security. A novel identity mapping and discovery system based on DHT-based directory service is proposed in [4]. This system enables users of web-based communication applications to discover and authenticate other users in the network. In [16], a mirror-presence mechanism is used to locate, identify and authenticate users on web calling services. While all these solutions facilitate user authentication and identification in WebRTC, they do not provide a method that ensures the legitimacy of users. Therefore, new mechanisms are still needed to screen and scrutinize callers over web communication services.

In order to anticipate user behavior, reputation-based techniques are one the most practical and effective solutions used over the Internet. Most of the reputation-based models used over P2P networks leverage on the collection of recommendations about the trustworthiness of a peer by other members of the network. The most popular reputation-based trust models for P2P networks include: EigenTrust [17], PeerTrust [18] and PowerTrust [19]. A comprehensive comparison of these models is provided in [20]. EigenTrust is one of the most well-known and cited trust model for P2P file sharing networks. PowerTrust is considered as an enhancement of EigenTrust. However, these models are based on the assumption of some pre-trusted peers in the network. In contrast, PeerTrust is a very simplistic model that determines peers trustworthiness by taking into account several important factors such as feedback source credibility, transaction context and community context.

On the other hand, reputation over on-line social networks is based on user interactions. Interaction-based trust models are usually applied to networks where the size, frequency and type of interaction are important indicators of trust. This is evident in the case of STrust [21], where trust is evaluated based on the popularity and engagement of users in social networks. A novel behavior-based trust model for on-line social networks is presented in [22]. These models completely ignore the structure of networks that provide important information about how members in a community relate to each other. In contrast, network-based trust models exploit the propagative nature of trust in the network to determine the trust between any two nodes. TidalTrust [23] provides a good illustration on how the network structure can be used to establish trust between peers having no direct connection.

In WebRTC, the CSP facilitates direct connection between communicating participants. However, the WebRTC based communication differs from P2P networks. P2P systems are usually deployed using a distributed hash table that allow peers to efficiently search the network for a resource. On the other hand, WebRTC standard requires a central server to discover and locate peers in order establish a communication session between them [8]. With WebRTC services, the centralized functionality of a server is used to maintain decentralized clusters of peers. Therefore, trust in WebRTC services needs to be computed in a centralized manner. The call graphs in communication services show how peers relate to each other. The frequency, duration and nature of their calls are

important behavioral indicators that can be used to estimate their trustworthiness [24]. We choose to explore this area of research to present the firs hybrid trust model for real-time web communication services.

III. THREAT TAXONOMY

In this section, we detail the threat taxonomy for real-time web conversation services, with a focus on social threats that directly target users.

The security threats in web communications are categorized into: confidentiality, integrity and social threats. Potential threats against user confidentiality includes unauthorized means of capturing information such as voice, data, identities, credentials and call patterns. Threats against integrity involve the alteration of signaling or media messages by intercepting them in the middle of the network. However, threats against social contexts are distinctive, as they are directly aimed against humans. In social threats, the context between communicating parties is manipulated in order to transfer false or malicious content to the target victim. Identity over web conversational services is commonly a combination of self-created user profiles and credentials. Therefore, an attacker can present fraudulent information, such as a false name, organization, email address, or presence information to misrepresent himself over the network. Identity misrepresentation over the telephone network facilitates various security threats.

We identify five social security threats that are present over web conversational services:

1) ***Phishing:*** Phishing is an illegal attempt to obtain some one's confidential information such as their identity, password, bank account number, credit card information etc. During a phone call, the attacker usually pretends to be from a trustworthy organization (such as a reputed bank or recognized office) in order to trick their victims in revealing private and confidential information.
2) ***Spam:*** Spam over Internet Telephony (SPIT) or robocalls are automatically dialed unsolicited pre-recorded bulk phone calls that are broadcasted for marketing purposes. SPIT are much more disruptive than other kinds of spam, as they require immediate response from the recipient. The low cost and open nature of Internet telephony provides an attractive medium for attackers to generate spam.
3) ***Undesired Content Distribution***: Communication services are used to distribute corrupted or virus-infected files such as spywares, viruses, trojans, malwares etc. Illegal and unlawful content may also be distributed such as sexually explicit images, content promoting crime or violence, copyright violation and illegal trading. This type of content can also be used to deliver false or misleading information, which can in turn be used for phishing attacks.
4) ***Nuisance Marketing***: Telemarketers use high pressure sales techniques over communication services to pursue customers in buying their products over phone calls. These advertising tactics are considered to be unethical. Telemarketers have also been involved in various frauds and scams while selling products over the phone.
5) ***Unwanted Contact***: Unwanted communication includes acts of harassment, extortion, blackmail and abuse that are all against the law. While a communication system may not be able to detect unwanted ot undesired communication, it may be able to detect the users involved in such activities.

IV. TRUST IN REAL-TIME WEB COMMUNICATION

WebRTC is being used as the underlying P2P technology to build novel web-centric communication platforms [2]–[6]. The vast adaption of the WebRTC standard for web calling relies on an efficient identity and trust management system. We elaborate on the necessity of computing trust in WebRTC based communication services.

There are two types of trust relationships in communication networks: (a) trust between a user and the CSP, and (b) trust between communicating participants. In the WebRTC standard, the CSP is responsible for the exchange of session parameters, identities, call answer/offer requests and user reachable addresses. Therefore, it is not possible to achieve user confidentiality from the CSP. However, trust between users and a CSP is established based on the validation of the ownership of website's origin. The digital certificate of the CSP is verified from the certificate issuing authority before its services are used.

In order to understand the notion of trust between communicating participants, we first need to describe the process of user identification in communication networks. Identities in traditional telco-operated networks are publicly known identifiers following the international telephone numbering system. They are not only used for authentication purposes but also for routing a call to the current location of a user's device. However, identities over the web are simply a combination of user's profile (name, age, email address etc) and credentials used for login purposes. Therefore, potential adopters of the WebRTC standard face two technical challenges related to user identity: i) user discovery and ii) identity provisioning.

User discovery involves an efficient identity resolution system that maps user identities to the currently available web address of user's device. This is essential in order to establish a session for the exchange of media. CSP is responsible for the process of signaling between communicating participants in order to establish a secure and reliable session. However, before establishing a connection it is necessary to ensure that users know who they are talking to. For this purpose, WebRTC architecture facilities end-to-end peer authentication using Identity Provider (IdP). IdP can be managed by the CSP itself or delegated to a trusted third party. End-to-end peer authentication requires the exchange and verification of user identities in a P2P fashion; this is known as identity provisioning.

Figure 1 presents a WebRTC call model where Alice and Bob are subscribers of 'TalkNow' service. They choose to identify themselves via their trusted IdPs. If Alice wants to talk to Bob, she uses the services of Talknow to discover Bob's web address of a currently available user device. Talknow is responsible for providing signaling between Alice and Bob in order to establish a communication session for the exchange of media. However, before establishing a session, Alice and Bob identify each other using IdP proxy mechanisms. The IdP-Proxy downlaoded from an IdP's URL provides an interface between IdP and browser for user authentication and verification purposes.

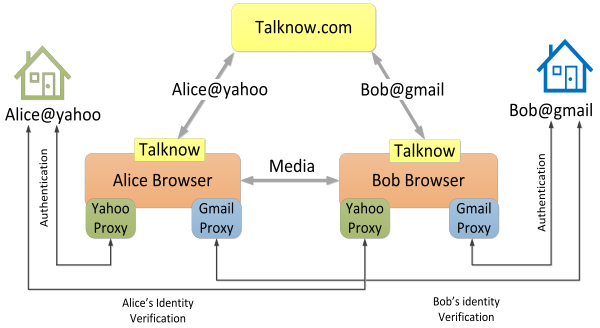


FIGURE 1. User identification in WebRTC call model.

Trust between communicating participants in WebRTC standard relies on mutual authentication. It is important to note that trust cannot be built merely on the basis of authentication. For example, if Bob is able to reliably and securely verify that Alice@gmail.com is owned by Alice, it does not mean that Bob can trust Alice. Mutual authentication allows communicating participants to identify each other but does not guarantee their trustworthiness. In order to determine trust between communicating participants methods of estimating trustworthiness and reputation should be built into web calling services.

V. TrustCall MODEL

In this section, we present '*TrustCall*', the first hybrid trust computational model that evaluates trust between communicating participants.

We define trust between communicating peers as the belief that they will act legitimately and securely over the communication session. Trust is dynamic in nature as it increases with positive experiences and decreases with negative ones. Trust should therefore be modeled with respect to time and expressed in a continuous variable. Since older experiences might become irrelevant with time, recent experiences are more important in determining trust. We consider $\hat{T}$ the time period over which communication occurs. The time period is further divided into n intervals, $[t_0, t_1]$, $[t_1, t_2]$, ..., $[t_{n-1}, t_n]$. A particular interval $[t_{k-1}, t_k]$, is referred to as the k^{th} interval, where for any interval $[t_i, t_j]$, $t_i < t_j$. Experiences that occur within the specific time period are weighted based on their positioning in time. Experiences outside the time period are ignored. The computed trust for a communicating peer p_i is denoted as:

$$Tr_t(p_i) \tag{1}$$

Figure 2 illustrates the *TrustCall* architecture comprised of three components: information collection, trust computation and trust dissemination.

A. INFORMATION COLLECTION

The information regarding user reputation is collected from two sources, (i) recommendations and (ii) user's communication behavior. Recommendations regarding user legitimacy are collected from other members of the network based on their experience. Each recommendation is selected and weighted based on the member's trustworthiness in give correct recommendations. In order to determine whether a communicating peer is worthy enough to be accepted as recommender, we introduce the social reliability parameter. Whereas, credibility parameter represents the sincerity of a recommender in giving correct recommendations. For each recommender facts about its social reliability and credibility are collected. On the other hand, a user's communication behavior is observed in order to determine their popularity and acceptance in the network. Three attributes of call graphs (incoming calls, outgoing calls and talk time) are observed to describe the behavior of a user in a communication network.

B. TRUST COMPUTATION

TrustCall is based on the evaluation of two types of trust: *Authenticity Trust* and *Behavioral Trust*. *Authenticity Trust* is used to describe the legitimacy of a communicating peer's identity. *Behavioral Trust* is computed to determine a user's acceptance and recognition by other members of the network. *TrustCall* is a hybrid trust model in which trust is expressed as a linear weighted sum of *Behavioral Trust* and *Authenticity Trust*. Each type of trust owns a weight that indicates its influence over the computed trust. The computed trust has a value between -1 and $+1$. This facilitates illustrating the amount of trust as well as distrust associated with any peer. The computed trust $Tr_t(p_i)$ for a peer p_i can be expressed as follows:

$$Tr_t(p_i) = \alpha \times Tr_{Auth}(p_i) + (1 - \alpha) \times Tr_{Beh}(p_i) \tag{2}$$

where $Tr_{Auth}(p_i)$ is the *Authenticity Trust* and $Tr_{Beh}(p_i)$ is the *Behavioral Trust*. α is the weight that ranges from $[0, 1]$. Quantifying the influence of each type of trust depends upon its usage in web communication networks.

C. TRUST USAGE

TrustCall allows CSPs to introduce the feature of trust visualization. Trust visualization is the presentation of trust in a pictorial, graphical or textual format. The visualization of a caller's reputation can be used to advise and assist whether and how much a particular peer can be trusted over the communication network. Any user will be able to visualize the

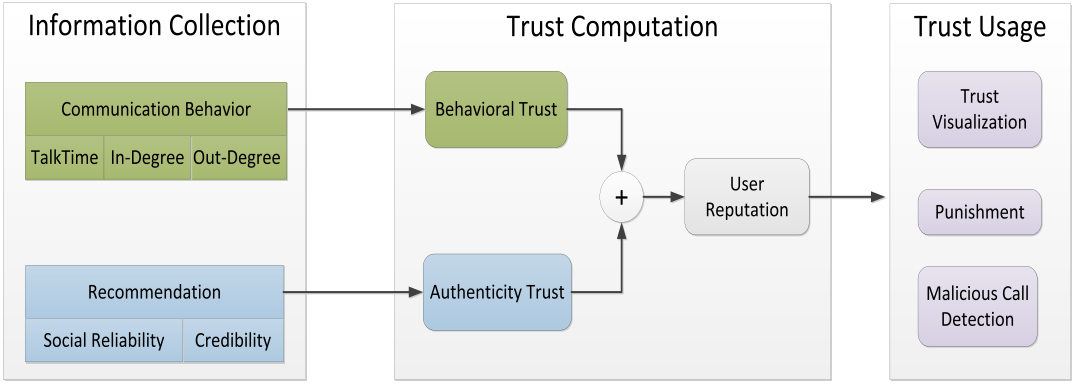


FIGURE 2. TrustCall architecture.

computed trust of other members of the network before initiating or accepting a call request. This will help subscribers to identify and communicate with legitimate callers. However, the decision to accept or reject a call request is very personal and is left up to the user to decide. The evaluated trust can further be used to enhance confidentially and security over a communication session. For example, a user may limit the amount of information or refrain from accepting any image or file from callers that are doubtful and suspicious. This trust value can also be used to block call requests that originate from the least-trusted members of the network. In addition to trust visualization, *TrustCall* allows CSPs to detect malicious callers over their networks. CSP may punish malicious callers by blocking their calls or by banning them from the network.

In *TrustCall* the reputation of a user is based on the computation of *Authenticity Trust* and *Behavioral Trust* as shown in Figure 2. We formally define *Authenticity Trust* in Section VI followed by *Behavioral Trust* in Section VII.

VI. AUTHENTICITY TRUST

Authenticity Trust describes the legitimacy and genuineness of user identity. *Authenticity Trust* of a peer is evaluated based on the recommendations received from its communicating participants. In *TrustCall* a recommendation is bound to each call where both participants rate each other based on their experiences. If the communicating participant has a genuine identity, it is rated as legitimate. If the communicating participant uses a false identity to conduct malicious activities as described in Section III, it is rated as malicious. Any peer p_j can rate its communicating participant p_i as follows:

$$Rec_{p_j \rightarrow p_i} = \begin{cases} +1 & \text{if legitimate} \\ -1 & \text{if malicious} \end{cases} \tag{3}$$

In traditional recommendation systems trust is commonly computed as the average aggregate of all recommendations received over a peer's communication lifespan. If n_{p_i} are the total number of communicating participants of peer p_i, then the *Conventional Trust* $Tr_{Conv}(p_i)$ can be computed as follows:

$$Tr_{Conv}(p_i) = \frac{\sum_{j=1}^{n_{p_i}} Rec_{p_j \rightarrow p_i}}{n_{p_i}} \tag{4}$$

Traditional recommendation systems are prone to several attacks and strategies used to unfairly enhance reputation [25]. In Table 1, we summarize adversarial powers that are accessible to malicious peers. For instance, peers may behave as a traitor, give false recommendations, conduct a Sybil attack, form a malicious collective group, or simply shed their bad reputation by re-entering the network with a new identity. Malicious peers adopt such strategies to avoid their detection in recommendation systems. In *Conventional Trust* recent ratings play an insignificant role in altering a peer's trust value. Moreover, each recommendation is considered equally to evaluate trust for peer p_i. Therefore, malicious peers can easily deceive or mislead traditional recommendation systems.

TABLE 1. Adversarial powers.

Behaviour	Description
Traitors	Traitors are users that behave properly for a period of time to maintain a respectable reputation before behaving maliciously.
Sybil Attack	Multiple false identities are forged by a user in order to enhance its reputation.
False Rating	Users may provide false recommendations. Malicious peers are more likely to provide false recommendations in order to hide their bad reputation.
Malicious Spies	Malicious spies are peers who behave legitimately in the network but give false rating to peers who behave maliciously.
Collusive Group	Peers in the network may form a collusive group in order to cooperate with each other by providing false rating.
White Washing	Users shed their bad reputations by purposely leaving and re-entering the network with a new identity.

In *Authenticity Trust* we introduce mechanisms to combat the attacks and strategies defined in Table 1. In order to capture peers recent behavior, we weight recommendations based on their positioning in time. This method helps in detecting traitors presence in the network. We model peer's

Authenticity Trust in terms of the number of recommendations received over n subintervals of a specified time period $\hat{T}$ (for instance 3 weeks or 3 months). The *Authenticity Trust* at time t_i is represented as follows:

$$\frac{\sum_{k=1}^{n} w_k \sum_{j=1}^{n_{p_i}^k} Rec_{p_j \rightarrow p_i}^k}{\sum_{k=1}^{n} n_{p_i}^k} \tag{5}$$

where n are the total number of subintervals of time period and $n_{p_i}^k$ are the total number of recommendations for p_i in k^{th} interval where $1 \leq k \leq n$. Each interval $[t_{k-1}, t_k]$ is weighted based on its position. Recommendations that occur in the older intervals of the time period are weighted less than the recommendations in recent intervals. Recommendations older than the specified time period are discarded. We use the position weight w_k defined in [26] for each interval, using $w_k = \frac{k}{S}$ where $S = \frac{n(n+1)}{2}$. The choice of time period $\hat{T}$ and number of intervals n is a matter of trust evaluation policy that is set by the CSP.

Furthermore, in *Authenticity Trust* we introduce two parameters i) social reliability and ii) credibility to choose and weight each recommendation received. Social reliability determines whether a communicating peer is worthy enough to be accepted as a recommender, while the credibility parameter represents the sincerity of a recommender in giving correct recommendations. In *Authenticity Trust* each recommendation received is weighted with both the social reliability and credibility parameters. The *Authenticity Trust* $Tr_{Auth_{t_i}}(p_i)$ of a peer p_i can be computed as follows:

$$Tr_{Auth_{t_i}}(p_i) = \frac{\sum_{k=1}^{n} w_k \sum_{j=1}^{n_{p_i}^k} Rec_{p_j \rightarrow p_i}^k \times Sr_{p_j} \times Cr(p_j)}{\sum_{k=1}^{n} n_{p_i}^k} \tag{6}$$

where Sr_p is the social reliability and $Cr(p_j)$ is the credibility of peer p_j.

Social reliability is a binary parameter that shows whether a user is reliable enough to be considered as a recommender. The recommendation from any peer is considered if its social reliability parameter is equal to 1. Social reliability is based on the number of interactions in the network. A peer's interaction rate is represented by the amount of calls made and received in the network. The thresholds can be set by examining the average number of incoming and outgoing calls in the network. Social reliability is introduced to detect fake profiles that are injected into the network. Fake profiles are highly unlikely to have a reasonable amount of interactions, as their sole purpose is to falsely recommend a particular peer. The social reliability parameter is defined as follows:

$$Sr_{p_j} = \begin{cases} 1 & \text{if } interactions \geq threshold \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

On the other hand, the credibility parameter helps in determining the sincerity of a peer in giving correct recommendations. It is the weight given to the recommendation based on the user's sincerity. The credibility parameter has a value between 0 and 1. Credibility close to 1 shows that the peer is sincere in giving correct recommendations, while credibility close to 0 shows that the peer provides false recommendations. Therefore, the credibility parameter helps to detect users who provide false ratings. Furthermore, it also helps to counter collusive group formation by selecting peers that provide correct recommendations. To determine user credibility we introduce three metrics: i) reliability, ii) similarity and iii) honesty.

A. RELIABILITY (R)

Reliability is based on the assumption that legitimate peers are more likely to give correct recommendations whereas malicious users are more likely to give false recommendation. Therefore, in the reliability metric we use the authenticity trust to determine a peer's credibility. Reliability at time t_i is determined using the authenticity parameter in the following manner:

$$Reliability_{t_i} = \begin{cases} Auth_{t_{i-1}} & \text{if } Auth_{t_{i-1}} \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

B. SIMILARITY (S)

This metric is based on the assumption that a legitimate user is more likely to communicate with legitimate peers over the network. The similarity metric measures the similarity of each peer with its neighbors (communicating participants) in terms of similar recommendations. Therefore, legitimate peers should have high similarity, whereas malicious users should have low similarity. To find the similarity for each peer p_j, a set of common peers that were rated by peer p_j and its neighboring peers are obtained. Similarity is then evaluated in the following manner:

$$Similarity = \frac{Number\ of\ Similar\ Recommendations}{Total\ Number\ of\ Recommendations} \tag{9}$$

C. HONESTY (H)

This metric indicates the honesty of a recommender by considering the degree to which the recommendations given by the peer are different from the evaluated authenticity trust. A recommendation provided at time t_i is considered as honest if its sign is the same as the sign of the evaluated *Authenticity Trust* Tr_{Auth} at t_{i-1}. Otherwise the recommendation provided is considered as a lie.

$$Honesty = \frac{Number\ of\ Honest\ Ratings}{Total\ number\ of\ Ratings} \tag{10}$$

VII. BEHAVIORAL TRUST

Behavioral Trust describes the trustworthiness of a user based on its popularity in the communication network. Popularity is an important indicator that illustrates user recognition and acceptance by other members in the network. The behavior of a user provides important information that can be used to determine its popularity. We use three basic attributes of call graphs to describe the communication behavior of a user:

A. TALK TIME

Talk time (Tk) is the total duration of the calls placed between two participants. The average talk time of a user is the total duration of calls divided by the number of calls placed or received. The frequency and duration of calls are important aspects that define trust relationship between two peers. Repeated calls and long call duration implies that peers have a strong trust relationship. Malicious peers usually have low average talk time as the called party tries to end the communication shortly after noticing malicious behavior [27]. On the other hand, legitimate peers are more likely to have a respectable average talk time as they are expected to have strong trust relationships with at least a few of their communicating participants [28].

B. IN-DEGREE

In a call graph of a communication network the in-degree value represents the number of calls received by a user. In-degree is an important attribute that can be used to determine the acceptance of a user within a network. Malicious peers are unlikely to have high in-degree values as they are the least popular members of the network. However, in-degree value alone cannot be used to determine the popularity of a peer, as the number of incoming calls also depends upon a caller's profile. For example, a travel agent will always receive a high number of in-coming calls compared to am accountant.

C. OUT-DEGREE

The out-degree of a user in a call graph represents the number of calls made by the user to other members of the network. Malicious peers are expected to have a high number of outgoing calls as the nature of their activities (spam, malicious content distribution and phishing) requires them to make a high number of outgoing call requests.

These attributes can be used to describe the behavior of users in a network. We intend to use these attributes to rank and categorize peers in a communication network. Malicious peers are characterized by their high number of outgoing calls, low number of in-coming calls and low average call duration [27]. On the other hand, legitimate peers usually tend to have high in-degree values and significant talk time [24].

Ranking algorithms are used in social networks to rank nodes using link analysis. The famous PageRank algorithm [29] determines the importance of a node based on the number of incoming links. We use the PageRank algorithm to rank communicating peers using their in-degree values. The incoming links are weighted by the talk time between two peers. This ensures that more importance is given to the links that have longer talk time. We define the page rank PR of communicating peer p_i using its in-degree and talk time as follows:

$$PR(p_i) = \frac{1-d}{N} + d \times \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)} \times Tk(p_i, p_j) \tag{11}$$

where $M(p_i)$ are a set of peers that link to peer p_i, and $L(p_j)$ are the number of outgoing links of peer p_j. d is the damping factor. In order to consider the outgoing links we use the inverse PageRank algorithm. In inverse PageRank the nodes having high number of outgoing links are ranked the lowest. By weighting the links with their talk time $Tk(p_i, p_j)$, less importance is given to the links that have low call duration. We define the inverse page rank PR' of peer p_i using its out-degree and talk time as follows:

$$PR'(p_i) = \frac{1-d}{N} + d \times \sum_{p_j \in M'(p_i)} \frac{PR'(p_j)}{L'(p_j)} \times Tk(p_i, p_j) \tag{12}$$

where $M'(p_i)$ is the set of peers that p_i links to and $L'(p_j)$ are the number of the incoming links of node p_j. In order to consider both incoming and outgoing links we introduce *RankCall* $RC(p_i)$ algorithm which aggregates PageRank and inverse PageRank as follows:

$$RC(p_i) = \frac{1-d_{f_i}-d_{f_o}}{N} + d_{f_i} \times \sum_{p_j \in M(p_i)} \frac{RC(p_j)}{L(p_j)} \times Tk(p_i, p_j) + d_{f_o} \times \sum_{p_j \in M'(p_i)} \frac{RC(p_j)}{L'(p_j)} \times Tk(p_i, p_j) \tag{13}$$

where d_{f_i} and d_{f_o} are the incoming damping factor and outgoing damping factor respectively. In order to ensure the convergence of the algorithm, we use $d_{f_i} = 0.85$ and $d_{f_o} = 0.25$ as noted in the Symrank algorithm [28]. Symrank algorithm uses in-degree and out-degree values to detect SPIT. However, it does not consider the talk time between communicating peers.

The rank defined by *RankCall* algorithm describes the popularity of a peer in the network. This rank can further be used to categorize peers in order to assign them trust values. This process is illustrated in the example below:

1) EXAMPLE

A CSP uses *RankCall* algorithm to categorize its subscribers into sets of highly popular, popular, neutral, unpopular and highly unpopular peers. Various ranking thresholds are set by the CSP in order to categorize peers. The ranking thresholds depend upon the network characteristics such as average talk time etc. The rank of a peer in the network can be used to determine its popularity as follows:

$$Pop(p_i) = \begin{cases} \text{Highly Popular} & \text{if } Rank_{th1} \leqslant RC < Rank_{th2} \\ \text{Popular} & \text{elseif } Rank_{th2} \leqslant RC < Rank_{th3} \\ \text{Neutral} & \text{elseif } Rank_{th3} \leqslant RC < Rank_{th4} \\ \text{Unpopular} & \text{elseif } Rank_{th4} \leqslant RC < Rank_{th5} \\ \text{Highly Unpopular} & \text{elseif } Rank_{th5} \leqslant RC \leqslant Rank_{th6} \end{cases} \tag{14}$$

The popularity of a peer p_i is further used to assign *Behavioral Trust*:

$$Tr_{Bev}(p_i) = \begin{cases} +1 & \text{if Highly Popular} \\ +0.5 & \text{if Popular} \\ 0 & \text{if Neutral} \\ -0.5 & \text{if Unpoular} \\ -1 & \text{if Highly Unpopular} \end{cases} \quad (15)$$

VIII. EXPERIMENTS AND RESULTS

The performance of *TrustCall* is analyzed in terms of *Authenticity Trust* and *Behavioral Trust* in this section. A network of communicating peers was generated to test the feasibility and effectiveness of our proposed trust model. Firstly, we show the effectiveness of *Authenticity Trust* against the various types of adversaries that prevail in recommendation systems. Secondly, we demonstrate *Behavioral Trust* by categorizing peers based on their popularity in the network. Lastly, the performance of the *TrustCall* model is compared with that of the *PeerTrust* model.

A. EXPERIMENTAL SETUP

We generated a network of communicating peers to test the feasibility and effectiveness of *TrustCall*. Table 2 summarizes the main parameters of the network. The structural properties of telecom call graphs [30] were used to incorporate the real characteristics of a communication network. In order to simulate a call graph, the BarabasiAlbert algorithm [31] was used to generate a random scale-free network of 300 communicating peers. The network was generated using 20 initially connected peers. New peers connect to existing peers with a probability proportional to their communication links. The BarabasiAlbert model uses a preferential attachment mechanism in which new peers introduced into the network prefer to communicate with already heavily linked peers. Therefore, the degree distribution of the network follows a power law distribution. The in-degree and out-degree power law exponents of the network is between $1.5 < \gamma < 2.5$, whereas the clustering coefficient of the network is between $0.75 - 0.8$.

TABLE 2. Communication network parameters.

Notation	Description	Value
N	Number of Communicating Peers in the Network	300
m_0	number of initially connected peers	20
γ	Power Law Exponent	1.5-2.5
C	Clustering Coefficient	0.75-0.8
n_{exp}	# of experiments over results are averaged	5
Tk_L	Talk time of a legitimate peer (sec)	124-204
Tk_M	Talk time of a malicious peer (sec)	$\leqslant 20$
n	Number of intervals	7

The communicating peers in the network are divided into two sets: Legitimate peers and Malicious peers. The synthetic call workload from [24] is used to set the duration of communication between participants. The call duration between communicating participants is generated using a normal distribution. The talk-time of calls originated by legitimate peers usually are between $124 - 204$ seconds whereas the talk-time of calls originated by malicious peers are generally less than 20 seconds. In the recommendation system built over the communication network we consider two assumptions: legitimate peers provide correct ratings, and malicious peers provide false ratings. The second assumption is generally true as malicious peers usually tend to give false ratings in order to hide their malicious behavior [18]. However, the first assumption may not necessarily be true as legitimate peers may provide false rating. Therefore, legitimate peers are considered to rate correctly with a probability of 0.8. On the other hand, malicious peers always rate other legitimate peers falsely. In the case of collusive group formation malicious peers cooperate with each other in order to enhance their reputations by rating each other falsely.

TrustCall recommends whether a caller is trustworthy or untrustworthy in order to differentiate between malicious and legitimate peers. If the computed trust of the caller is greater than 0, the caller is considered as trustworthy, otherwise it is considered untrustworthy. The decision whether to communicate or not is very personal and is left up to the user to decide. For experimentation purposes, we consider that a user always rejects calls originating from untrustworthy callers and accepts calls originated by trustworthy callers. We used two performance metrics to demonstrate the efficiency of our proposed model:

1) TRUST COMPUTATION ERROR

Trust computation error is the total number of errors occurred divided by the total number of communicating peers. An error occurs when *TrustCall* declares a malicious peer as trustworthy or declares a legitimate peer as untrustworthy.

2) USER SATISFACTION

User satisfaction in the network is the overall number of satisfied call transactions divided by the total number of calls placed within the network. Users are considered satisfied when they accept a legitimate call or reject a malicious call.

B. PERFORMANCE EVALUATION OF AUTHENTICITY TRUST

Authenticity Trust evaluates the genuineness and legitimacy of a user's identity. It is based on recommendations received from other members of the network. However, recommendation systems are vulnerable to several threats and adversaries. Therefore, we evaluate the effectiveness and robustness of *Authenticity Trust* against typical adversaries present in the recommendation system. We conducted four different experiments to demonstrate the performance of *Authenticity Trust* in the presence of traitors, false ratings, Sybil attacks and collusive groups. The objective of these experiments is to evaluate the robustness of the *TrustCall* model against different behaviors of malicious peers. Therefore, in each experiment we tested our model such that malicious nodes make up between 0% and 100% of all nodes in the network.

Experiment 1: The first experiment shows the effectiveness of *Authenticity Trust* against traitors. In this experiment,

we consider a time period divided into 7 equal subintervals over which calls are placed. The trust computation error is computed against an increasing number of malicious peers. We consider 50% of the malicious peers present in the network as traitors. Traitors behave legitimately in the initial subintervals of the time period to earn good reputation after which they start acting maliciously. The other 50% behave maliciously throughout the time period.

Analysis: Figure 3 compares the performance of *Authenticity Trust* described by Equation 4 with *Conventional Trust* represented by Equation 5. It can be observed that the performance of the conventional approach decreases significantly as the number of malicious peers in the network increases. This is due the presence of traitors which remain undetected in the conventional approach. In *Conventional Trust* recent ratings play an insignificant role in altering a peer's trust value. Therefore, traitors can maintain a respectable reputation value by shifting their behaviors. However, in *Authenticity Trust* each rating is weighted based on their positioning in time. Recent ratings are considered more important than old ratings and ratings beyond a specific time period are discarded. This dynamic evaluation allows to detect the behavior of traitors in the network.

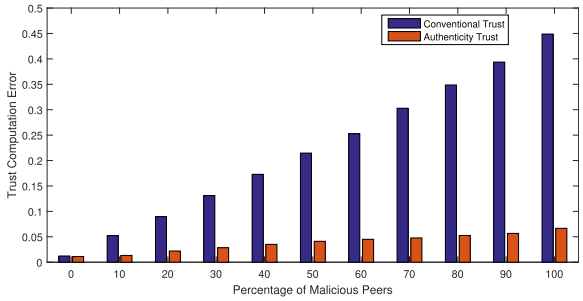


FIGURE 3. Trust computation error in the presence of traitors.

Conclusion: Authenticity Trust performs consistently over an increasing number of malicious peers in the network. The dynamic evaluation of *TrustCall* provides an effective mechanism against traitors present in the network. However, peers who consistently behave in an acceptable manner but decide to act maliciously once in a while will still remain undetected.

Experiment 2: The second experiment was conducted to show the robustness of *Authenticity Trust* against Sybil attacks. In this experiment, we consider that 50% of the malicious peers present in the network will carry out a Sybil attack. A Sybil attack is conducted by creating and introducing 30 fake peers in the network. Therefore, for each malicious peer conducting a Sybil attack we inject 30 fake peers into the network. These peers communicate and provide false rating to the user conducting the Sybil attack. As their sole purpose is to enhance user's reputation, such peers have fairly low interaction rate in the network.

Analysis: In Figure 4 we compare *Authenticity Trust* with *Conventional Trust*. It can be observed that *Authenticity*

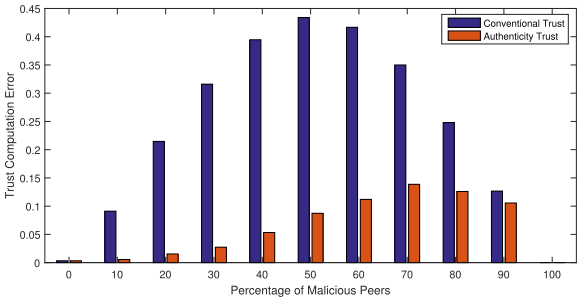


FIGURE 4. Trust computation error in the presence of Sybil attack.

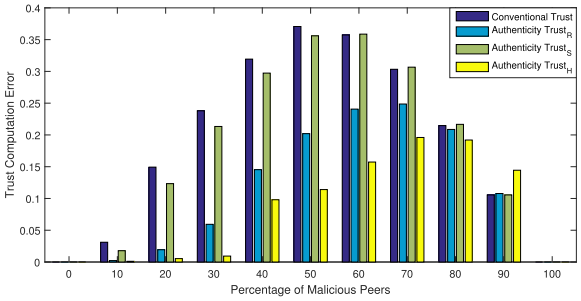


FIGURE 5. Performance of TrustCall in the presence of malicious peers.

Trust provides an effective defense mechanism against Sybil attacks. This is due to the fact that it relies on Social Reliability parameter described by Equation 7 to combat Sybil attacks. Social Reliability parameter allows *Authenticity Trust* to consider ratings received from socially reliable peers. In this experiment, we consider a peer to be socially reliable if it has in-degree higher than 5. This threshold is selected based on average in-degree value. Ratings from all other peers are rejected as they are likely to be given by fake peers introduced to conduct a Sybil attack. However, this also leads to rejection of rating coming from legitimate peers who have low interaction rate in the network. This experiment does not consider the presence of traitors and collusive groups in the network. Therefore, no error is observed when all peers in the network are malicious in nature.

Conclusion: Social Reliability parameter provides an effective defense mechanism against Sybil attacks by selecting socially reliable peers.

Experiment 3: This experiment compares the three metrics used to compute the credibility of peers. We recall that the credibility parameter determines the ability of a peer to give true recommendations. Three metrics are used to determine user credibility: reliability, similarity and honesty expressed by Equation 8, 9 and 10 respectively. In this experiment, cooperation between malicious peers is not considered. Thus the network considered is non-collusive in nature. A malicious peer present in the network rate other malicious peers correctly whereas rate legitimate peers falsely.

Analysis: Figure 5 compares the performance of *Conventional Trust* with that of *Authenticity Trust* in terms of trust computation error. We observe that the conventional approach

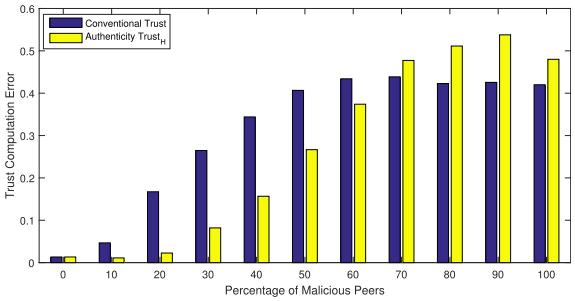


FIGURE 6. Trust computation error in the presence of collusive grouping.

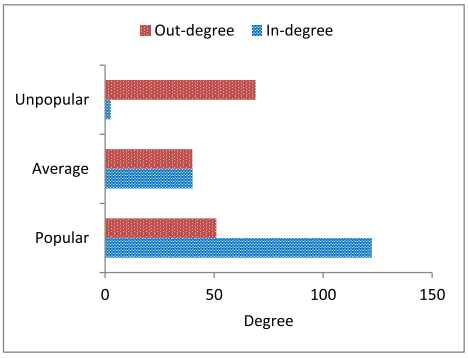
is very sensitive to peers who provide false recommendations. This is due to the fact that all recommendations received are considered equally. On the other hand, in *Authenticity Trust* each recommendation is weighted with the credibility of the user. We can observe that using the reliability metric in *Authenticity Trust*$_R$ the false recommendations can be considerably filtered out. However, the similarity metric in *Authenticity Trust*$_S$ is not very effective in estimating a peer's credibility. This was attributed to our setup with a highly clustered network in which a large number of calls were placed between malicious and legitimate peers. The Honesty metric in *Authenticity Trust*$_H$ shows the best results in the presence of false ratings.

Conclusion: Honesty metric is able to detect liars present in the network by comparing each rating with the computed trust. Thus each peer's recommendation is weighted with the ability of that peer to lie in the network. In case of very large number of liars present in the network this metric may not perform adequately as computed trust would largely be based on false recommendations.

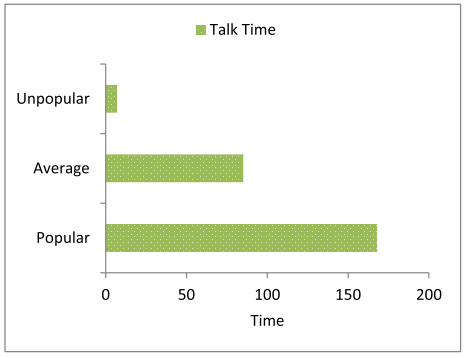
Experiment 4: Lastly, the feasibility of *Authenticity Trust* is tested under attack by collusive groups formed by malicious peers in the network. In a collusive group, malicious peers cooperate with each other by rating each other falsely, thereby enhancing their reputation in the network. In this experiment, we divide malicious peers into two sets of collusive groups. Malicious peers cooperate with each other by giving false ratings to each other inside the group. However, outside their group they rate correctly. We choose to examine *Authenticity Trust*$_H$ in the presence of collusive groups as honesty metric performs best in determining a peer's credibility.

Analysis: Figure 6 shows that the *Authenticity Trust* performs very well in the presence of collusive groups. The performance worsens when the percentage of malicious peers is very high. This is because in large collusive groups a high number of malicious peers cooperate with each other. The evaluated trust is largely based on false ratings and it is difficult for *TrustCall* to detect liars in the network.

Conclusion: Authenticity Trust performs very well in the presence of collusive groups. A very large collusive group is unlikely to occur in a communication network. A high number of smaller disjointed collusive groups may be present.



(a)



(b)

FIGURE 7. Performance of all peers with 10% highest and lowest ranked peers. (a) Mean degree of the network. (b) Mean talk time of the network.

C. PERFORMANCE EVALUATION OF BEHAVIORAL TRUST

Behavioral Trust is used to describe the popularity and acceptance of a user in a communication network. The *Rank Call* algorithm described by eq 13 is used to rank and classify peers based on their in-degree, out-degree and talk time values. This classification is further used to assign them trust values. Therefore, in this experiment we show the performance of our algorithm that assigns *Behavioral Trust* values in *TrustCall* model.

Experiment 5: We consider a CSP having 300 subscribers, where 25% of its subscribers are malicious in nature. The CSP uses *RankCall* to rank callers and categorizes them into different popularity sets as described by Equation 14. In this experiment we illustrate the average behavior of all peers in the network compared with the popular and unpopular ranked peers in the network. The 10% of the highest-ranked peers are declared as popular whereas 10% of the lowest-ranked peers are declared as unpopular.

Analysis: Figure 7a illustrates the mean degree of the peers communicating in the network. We can observe that the in-degree of popular peers is much more than the average in-degree of the network. This shows their importance and acceptance in the network. On the other hand, the in-degree of the lowest-ranked peers is very low compared to the average in-degree value of the network. Furthermore, their out-degree value is almost 10 times that of their in-degree value. Thus, they are likely to be involved in malicious activities such as

spam, phishing and nuisance marketing. As they are highly unpopular due to their involvement in malicious activities, they do not receive a lot of incoming calls from other members of the network. We can also observe from Figure 7a that the call duration for the lowest ranked peers is much lower than the average talk time of the network. Call duration is an important metric to define trust relationships between peers. Therefore, higher ranked peers should be assigned with high trust values, while low-ranked peers should be assigned with low trust values as they are the least trusted in the network.

Conclusion: This experiment proves that peers with low in-degree, low call duration and high out-degree values are ranked the lowest by the *RankCall* algorithm. Such peers are much more likely to be malicious in nature. Therefore, *Behavioral Trust* assigns them low trust values as described in Equation 15. Callers who have high in-degree and high call duration values are ranked the highest and thus will be more likely to act legitimately in the network. The behavior patterns of each caller vary based on different attributes of their profile such as geographical location, profession and interests. As our future work, we plan to study the behavior of different types of users present in the communications network to enhance mechanisms in order to evaluate *Behavioral Trust*. Furthermore, we intend to use behavior patterns to differentiate between different types of malicious behaviors such as fake profiles, spam, phishing etc.

D. EFFECTIVENESS OF TrustCall

In this subsection, we compare our approach with one of the most simplistic yet effective trust mechanisms, known as 'PeerTrust'. Peertrust is a reputation based trust model used over P2P file sharing networks. It considers various factors to quantify the trustworthiness of users over P2P networks [18]. PeerTrust provides a reasonably good performance against oscillating behaviors, collusive groups, false ratings, and man-in-the-middle attacks in reputation systems. The performance of PeerTrust is equivalent to that of other popular P2P trust models such as EigenTrust and PowerTrust [20]. However, it is considered to be the most simple and easy to implement trust model. Therefore, we chose to compare the performance of *TrustCall* with *PeerTrust* when applied over web conversational services. However *PeerTrust* only considers recommendations to evaluate user reputation. Therefore, in order to have a fair comparison we chose $\alpha = 1$ for *TrustCall* model in eq 2.

Experiment 6: We compare *TrustCall* with *PeerTrust* under three scenarios i) collusive ii) non-collusive and iii) Sybil attacks. The network settings for collusive, non-collusive and Sybil attacks are same as those specified in Section VIII-B. We compare user satisfaction when no trust is computed and when trust is computed using *TrustCall* and *PeerTrust*.

Analysis: Figure 8 provides the performance comparison in terms of user satisfaction with respect to the number of malicious peers present in the network. If trust is not computed all calls are accepted whether they are malicious or legitimate in nature. On the other hand, trust computation allows users to decide whether to accept or reject calls based on a caller's trustworthiness. Trust is computed using *PeerTrust* and *TrustCall* models. We can observe from figure 8 that there is a linear decrease in the performance when no trust is computed. However, user satisfaction improves immensely when the caller's trustworthiness is computed.

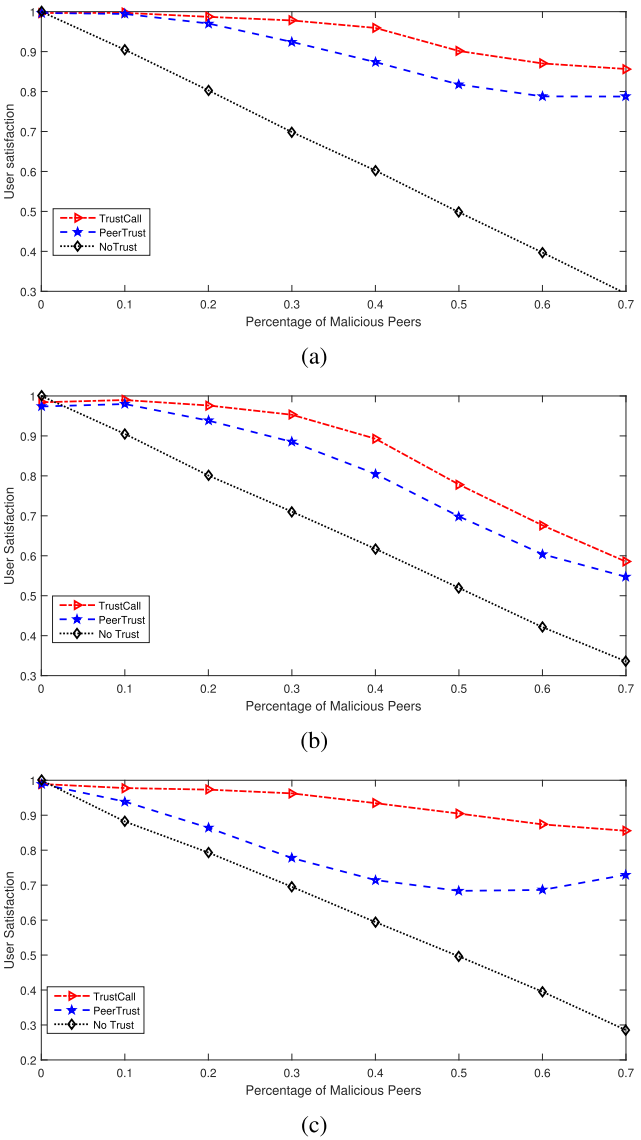


FIGURE 8. Performance Comparison in Non-Collusive, Collusive and Sybil attack. (a) User satisfaction in non-collusive network. (b) User satisfaction in collusive network. (c) User satisfaction under Sybil attacks.

Conclusion: TrustCall outperforms *PeerTrust* in all three scenarios. Thus, proving its effectiveness over communication networks. The trust computed by *TrustCall* is better suited to distinguish between malicious and legitimate users present in communication networks. However, peers may be able to discard their bad reputation by re-entering the network with a new identity. To completely prevent users from re-entering the network, new methods of identity verification should to be introduced that would restrict users ability to make duplicate identities over communication networks.

IX. CONCLUSION

Nuisance calls continue to be very disruptive and insecure in nature. Web conversational services remain an attractive medium for attackers to generate spam, conduct phishing and distribute malicious content. Therefore, we present '*TrustCall*' a simplistic heuristic-based trust model that computes reputation of callers in web communication networks. To the best of our knowledge, this is the first ever attempt to estimate trust in real-time web communications. In order to demonstrate the effectiveness and robustness of our approach we have reported several simulation-based experiments. Our research on computing trust in real-time web communications continues along several directions. Firstly, we aim to use communication behaviors to identify and differentiate between different types of malicious peers present in communication networks. Secondly, we intend to extend *TrustCall* for inter-domain communication frameworks where users from different service providers are able to communicate with each other. Furthermore, we plan to provide trust visualizations for real-time web communications in order to efficiently demonstrate the trustworthiness of callers in web communication networks.

ACKNOWLEDGMENT

The authors express our appreciation to Ahmed Bouabdallah, Assistant Professor, IMT Atlantique for his valuable comments on defining threat taxonomy which greatly improved the manuscript.

REFERENCES

[1] A. Bergkvist, D. C. Burnett, C. Jennings, A. Narayanan, and B. Aboba, "WebRTC 1.0: Real-time communication between browsers," W3C, W3C Editor's Draft, Tech. Rep., May 2016.

[2] S. Becot, E. Bertin, J. M. Crom, V. Frey, and S. Tuffin, "Communication services in the Web era: How can telco join the ott hangout?" in *Proc. 18th Int. Conf. Intell. Next Generat. Netw.*, Feb. 2015, pp. 208–215.

[3] E. Bertin, S. Cubaud, S. Tuffin, N. Crespi, and V. Beltran, "WebRTC, the day after: What's next for conversational services?" in *Proc. 17th Int. Conf. Intell. Next Generat. Netw. (ICIN)*, Oct. 2013, pp. 46–52.

[4] I. Friese *et al.*, "Cross-domain discovery of communication peers identity mapping and discovery services," in *Proc. Eur. Conf. Netw. Commun. (EUCNC)*, Jun. 2017, pp. 451–456.

[5] P. Chainho *et al.*, "Decentralized communications: Trustworthy interoperability in peer-to-peer networks," in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2017, pp. 1–5.

[6] I. T. Javed *et al.*, "Cross-domain identity and discovery framework for Web calling services," *Ann. Telecommun.*, vol. 72, nos. 7–8, pp. 459–468, 2017.

[7] A. D. Keromytis, "Voice-over-ip security: Research and practice," *IEEE Security Privacy*, vol. 8, no. 2, pp. 76–78, Mar. 2010.

[8] E. Rescorla, "WebRTC security architecture," IETF, IETF Internet Draft, Fremont, CA, USA, Tech. Rep., Jun. 2016.

[9] V. Beltran, E. Bertin, and N. Crespi, "User identity for WebRTC services: A matter of trust," *IEEE Internet Comput.*, vol. 18, no. 6, pp. 18–25, Nov. 2014.

[10] W. De Groef, D. Subramanian, M. Johns, F. Piessens, and L. Desmet, "Ensuring endpoint authenticity in WebRTC peer-to-peer communication," in *Proc. 31st Annu. ACM Symp. Appl. Comput.*, New York, NY, USA, 2016, pp. 2103–2110.

[11] E. Rescorla, "Security considerations for WebRTC," IETF Internet Draft, Fremont, CA, USA, Tech. Rep., Jul. 2013.

[12] S. Loreto and S. P. Romano, "Real-time communications in the Web: Issues, achievements, and ongoing standardization efforts," *IEEE Internet Comput.*, vol. 16, no. 5, pp. 68–73, Sep. 2012.

[13] V. Beltran, E. Bertin, and S. Cazeaux, "Additional use-cases and requirements for WebRTC identity architecture," IETF, Internet-Draft, Fremont, CA, USA, Tech. Rep., Mar. 2015.

[14] R. Copeland, K. Corre, I. Friese, and S. E. Jaouhari, "Requirements for trust and privacy in WebRTC peer-to-peer authentication," IETF, Internet-Draft, Fremont, CA, USA, Tech. Rep., Sep. 2016.

[15] L. Lopez-Fernandez *et al.*, "Authentication, authorization, and accounting in WebRTC paas infrastructures: The case of Kurento," *IEEE Internet Comput.*, vol. 18, no. 6, pp. 34–40, Nov. 2014.

[16] L. Li, W. Chou, Z. Qiu, and T. Cai, "Who is calling which page on the Web?" *IEEE Internet Comput.*, vol. 18, no. 6, pp. 26–33, Nov. 2014.

[17] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The Eigentrust algorithm for reputation management in p2p networks," in *Proc. 12th Int. Conf. World Wide Web*, New York, NY, USA, 2003, pp. 640–651

[18] L. Xiong and L. Liu, "PeerTrust: Supporting reputation-based trust for peer-to-peer electronic communities," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 7, pp. 843–857, Jul. 2004.

[19] R. Zhou and K. Hwang, "PowerTrust: A robust and scalable reputation system for trusted peer-to-peer computing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 18, no. 4, pp. 460–473, Apr. 2007.

[20] F. G. Mármol and G. M. Pérez, "Security threats scenarios in trust and reputation models for distributed systems," *Comput. Secur.*, vol. 28, no. 7, pp. 545–556, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404809000534

[21] S. Nepal, W. Sherchan, and C. Paris, "STrust: A trust model for social networks," in *Proc. IEEE 10th Int. Conf. Trust, Secur. Privacy Comput. Commun. (TrustCom)*, Nov. 2011, pp. 841–846.

[22] S. Adali *et al.*, "Measuring behavioral trust in social networks," in *Proc. IEEE Int. Conf. Intell. Secur. Inform.*, May 2010, pp. 150–152.

[23] J. Golbeck, "Personalizing applications through integration of inferred trust values in semantic Web-based social networks," in *Proc. 4th Int. Semantic Web Conf. Semantic Netw. Anal. Workshop*, Nov. 2005, p. 30.

[24] N. Chaisamran, T. Okuda, G. Blanc, and S. Yamaguchi, "Trust-based VoIP spam detection based on call duration and human relationships," in *Proc. IEEE/IPSJ Int. Symp. Appl. Internet*, Jul. 2011, pp. 451–456.

[25] S. Marti and H. Garcia-Molina, "Taxonomy of trust: Categorizing P2P reputation systems," *Comput. Netw.*, vol. 50, no. 4, pp. 472–484, 2006.

[26] I. Ray and S. Chakraborty, "A vector model of trust for developing trustworthy systems," in *Computer Security-ESORICS*. Berlin, Germany: Springer, 2004, pp. 260–275.

[27] J. Quittek, S. Niccolini, S. Tartarelli, and R. Schlegel, "On spam over Internet telephony (SPIT) prevention," *IEEE Commun. Mag.*, vol. 46, no. 8, pp. 80–86, Aug. 2008.

[28] H. K. Bokharaei, A. Sahraei, Y. Ganjali, R. Keralapura, and A. Nucci, "You can SPIT, but you can't hide: Spammer identification in telephony networks," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 41–45.

[29] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the Web," Stanford InfoLab, Stanford, CA, USA, Tech. Rep. 1999-66, 1999.

[30] A. A. Nanavati *et al.*, "On the structural properties of massive telecom call graphs: Findings and implications," in *Proc. 15th ACM Int. Conf. Inf. Knowl. Manage.*, 2006, pp. 435–444.

[31] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Rev. Modern Phys.*, vol. 74, pp. 47–97, Jan. 2002.

IBRAHIM TARIQ JAVED received the B.Sc. degree in telecommunication engineering from the National University of Computer and Emerging sciences (NUCES-FAST) and the master's (by research) degree in electrical engineering from the Lahore University of Management Science (LUMS-SBASSE). He is currently pursuing the Ph.D. degree with the Service Architecture Laboratory, Institut Mines-Telecom, Telecom SudParis, France. He has several years of research experience at various academic institutes. He is also a Researcher with the Service Architecture Laboratory, Institut Mines-Telecom, Telecom SudParis. His research interests include identity and trust management, real-time Web communications, peer-to-peer communications, channel modeling, and nanonetworks.

KHALIFA TOUMI received the master's degree from the National School of Computer Science in 2010 and the Ph.D. degree in computer science from Telecom and Management SudParis, France, in 2014. He is currently pursuing the degree in computer engineering with the National School of Computer Science, Tunisia. He was with SAGEMCOM as a Software Engineer for one year. He is also a Research Development Engineer with the CNRS SAMOVAR Lab, Telecom SudParis. He was involved in several European and French research projects. His topics of interest cover security testing and monitoring, trust management, and security policies specification for distributed systems.

NOEL CRESPI received the master's degrees from the University of Orsay (Paris 11) and the University of Kent, U.K., the Diplome d'Ingenieur degree from Telecom ParisTech, and the Ph.D. and Habilitation degrees from Paris VI University (Paris-Sorbonne). Since 1993, he has been with CLIP, Bouygues Telecom, and then with Orange Labs in 1995. He took leading roles in the creation of new services with the successful conception and launch of Orange prepaid service, and in standardization (from rapporteurship of IN standard to coordination of all mobile standards activities for Orange). In 1999, he joined Nortel Networks as a Telephony Program Manager, architecting core network products for EMEA region. In 2002, he joined the Institut Mines-Telecom and is currently a Professor and the Program Director, leading the Service Architecture Laboratory. He coordinates the standardization activities for the Institut Mines-Telecom, ITU-T, ETSI, and 3GPP. He is an Adjunct Professor with KAIST, an Affiliate Professor with Concordia University, and a Guest Researcher with the University of Goettingen. He is also the Scientific Director of the French-Korean Laboratory ILLUMINE. His current research interests are in sofwarization, data analysis, and Internet of Things/Services.