A Provider-Level Reputation System for Assessing the Quality of SPIT Mitigation Algorithms

Christoph Sorge and Jan Seedorf
NEC Laboratories Europe, Network Research Division
Heidelberg, Germany
Email: {christoph.sorge,jan.seedorf}@nw.neclab.eu

Abstract—The prevention of Spam over IP telephony (SPIT) is one of the greatest challenges for future large-scale deployments of VoIP telephony solutions. Some useful information for detecting SPIT calls is only available at the caller’s VoIP provider. Recent approaches therefore suggest the signalling of such information among providers. However, there is currently no way for a receiving provider to assess the trustworthiness or the semantics of SPIT-related information received. Our approach tackles this problem by applying a provider-level reputation system, based on SPIT tags assigned to outgoing SIP messages by the caller’s provider. The system provides an incentive to tag outgoing calls correctly, and it translates tags with arbitrary semantics into meaningful SPIT probabilities. We show analytically that our system significantly improves the receiving provider’s assessment of SPIT tags.

I. INTRODUCTION

Voice-over-IP (VoIP) telephony is rapidly gaining importance. As history has shown, the establishment of new, cheap means of communication inevitably bears the risk of being abused. Unless countermeasures are developed at an early stage, VoIP telephony may share the fate of the e-mail system, being mostly used for Spam messages. Various approaches have been developed in order to overcome the problem. Traditional filtering techniques on the recipient side, combined with advanced methods like the use of audio CAPTCHAs, are promising approaches. However, very important information about a caller (e.g., the rate at which he or she places outgoing calls) is available on the side of the originating SIP service provider (SSP). In today’s e-mail system, such information would not be worth much, as this system is very open and allows a mail transfer agent so send e-mails without proving its identity, and trust relationships between e-mail servers normally cannot be established. VoIP providers that directly exchange SIP messages, on the other hand, have normally established their identities, and they have contractual agreements. Also, SIP messages are usually exchanged securely (e.g., via TLS) [1], between providers. Hence, there is a unique opportunity to combine SPIT-related information available at different locations on the communication path in order to better mitigate the problem of unsolicited communication. To exploit this advantageous characteristic of interconnected VoIP networks, researchers have suggested sending SPIT scores from the sender’s domain to the receiver’s domain in-band, attached to a SIP message [2].

As a drawback, there is no way for the receiving SSP to interpret the SPIT-related information provided by the sender’s SSP regarding the likelihood of a message being SPIT. In addition, a provider may have flawed detection algorithms, suffer from configuration errors, or it might simply be incompetent to assess the SPIT likelihood for its outgoing calls. Further, even though there is a peering relationship between SSPs, a provider might be collaborating with spamming customers (or have compromised systems) and send out false information.

Therefore, some way should be found to decide whether or not a provider which attaches SPIT-related information to outgoing messages can be trusted to perform this tagging correctly. Such a decision has traditionally been the task of reputation systems. Our main contribution is the introduction of a provider-level reputation system for the task of SPIT prevention, which enables the SSP at the receiving end of the communication path to assess the quality of a sending SSP’s SPIT estimation algorithms. As a major advantage of our system, these algorithms can be proprietary and might even encompass non-disclosed functionality. The outcome of such a SPIT estimation algorithm can then be used to process the message further. By considering tuples (Sending provider’s identity, algorithm, score) as entities to be rated by the system, we also provide an easy way for a receiving SSP to interpret SPIT scores. Since a provider can reduce the amount of false positives by correctly tagging malicious outgoing calls, our system has the additional advantage of providing an incentive for correct tagging.

II. RELATED WORK

As the paper at hand suggests a reputation system for SPIT prevention, there are three categories of related work: work on reputation systems, on SPIT prevention, and on the use of reputation systems for SPIT prevention.

Reputation systems collect ratings from their participants, aggregate them, and make them available to their participants. Aggregation schemes used are often based (explicitly or implicitly) on a trust model. The eBay reputation system, being among the most prominent centralized approaches, has been analyzed by Resnick and Zeckhauser [3] (however, the system has been changed since then). Though distributed trust management has been discussed as early as 1996 [4], distributed reputation systems are a more recent development. They are mainly applied in Peer-to-Peer and in mobile ad-hoc networks. In both cases, the most important aims are to find useful resources, to reduce “free riding”, to privilege participants providing resources and to “punish” malicious
nodes. The Eigentrust system [5] computes a global trust value for each participant with a distributed algorithm. Other approaches for distributed reputation management have been suggested by Damiani [6] and Cornelli [7]. For an introduction and a comprehensive survey, see [8] and [9].

The topic of SPIT prevention is addressed in a number of research papers. SPIT prevention frameworks [10] determine how different detection modules can work together. Single detection algorithms, which may be used as modules in a SPIT prevention framework, exist in a considerable number already. They include approaches based on blacklists, whitelists, call rates, communication pattern analysis [11], and Turing tests. For a more comprehensive overview and a discussion of these methods, see [12] and [10].

Using reputation systems for SPIT prevention has only been studied recently, and by few authors. Rebahi and Sisalem [13] discuss SPIT detection using trust in social networks. Balasubramanyan et al. [14] introduce the CallRank system, in which both the local neighborhood (i.e., entities that have been called by the user) and global reputation values (computed using the Eigentrust system) are used, and only calls from reputed users are accepted. Kolan and Dantu [15] present a framework for SPIT detection based on the caller’s reputation. Patankar et al. [16] combine a reputation system with a social network of VoIP users. RFC 5039 [17] gives a general introduction to the problems of reputation systems and their specific drawbacks for the SPIT prevention problem, leading to the conclusion that only systems based on positive reputation can work.

Existing reputation-based SPIT prevention systems suffer from a number of problems, most of them caused by the fact that they work on the individual users’ level. Firstly, it seems likely that users may change their SIP identities (e.g., in order to get cheaper rates from different SIP service providers) quite frequently, leading to a loss of their acquired reputation. Secondly, the large number of users means that scalability becomes a vital issue. Thirdly, such reputation systems are a threat to users’ privacy. If two persons have trust relationships concerning their call behavior, this means that they phone each other. Information about such communication is legally protected in most countries throughout the world, including the EU member states\(^1\) and the U.S.\(^2\). Though these regulations would allow using user-level reputation systems with the users’ consent, a more privacy-friendly approach is desirable.

As a consequence, the reputation system we introduce in the paper at hand uses a different approach. Instead of trying to judge the trustworthiness of a user, we gather information about SSPs. In addition, we do not limit ourselves to checking whether a high number of SPIT calls originates from a certain provider. Instead, we extend the scope of the reputation system to the quality of the provider’s tagging of outgoing calls (if the provider decides to perform such a tagging, for which our reputation system provides a strong incentive).


\(^2\)U.S. Code, title 47, chapter 5, § 222

III. SYSTEM DESIGN

A. Assumptions and Design Considerations

Our design is based on a number of assumptions and preconditions which we believe to be realistic for future VoIP networks. First, SSPs using our reputation system must have a set of other providers which they trust to send correct information about the quality of received SPIT tags. Second, establishing reputation requires the rated entities to be identified—typically using certificates provided during the setup of TLS sessions. Identities should be long-lived; this can be a consequence of peering agreements, or encouraged by attributing the lowest possible reputation value to new identities. Third, there is a way for participating providers to determine if a call is SPIT, either during the call or after the call has terminated. For instance, a provider can assess if a call has been SPIT if callees provide feedback, e.g., by pushing a button on their phone [18]. Further, we assume that SSPs can communicate over secure (authenticated) channels.

We design a reputation system on the level of SIP service providers rather than individual users. This is not only for scalability reasons or due to an anticipated longevity of identities, but also because we want to exploit the service providers’ ability to observe their customers’ behavior. We opt for a distributed reputation system: it is unrealistic to assume that all SSPs worldwide can agree on a single reputation provider.

When assessing an entity’s reputation, it is important how to weight other entities’ ratings. Our system uses neither the reputation of the rating entity—which does not make a statement of its ability to judge others—nor its size. Instead, we base our reputation system on existing (or manually established) trust relationships between SSPs (see section III-B).

Reputation systems often try to calculate a “global”, unique reputation value for each rated entity. However, this implies a strong notion of transitive trust. As, based on the reputation system, legitimate calls might be rejected, we assume that SSP service providers will prefer having more control over their trust relationships, i.e. keeping the trust chains short. Furthermore, supporting long chains of transitive trust relationships requires using highly aggregated reputation data: the high number of (transitive) trust relationships means that no complete information can be sent to all trusted entities due to scalability considerations; instead, some of the information must be left out. Should this highly aggregated scheme turn out to be inadequate, its implementation must be changed on all participating systems. We therefore choose to support only very limited transitive trust, but to exchange only slightly aggregated data between nodes. This way, innovative SPIT detection algorithms can be easily deployed on the end points, rather than having to change the aggregation algorithms on intermediate nodes—a rationale very similar to the end-to-end argument of internet design.

B. Trust

SSPs have different kinds of trust relationships. Peering agreements represent trust that fees will be paid and that the
contract partner is not a malicious attacker. However, peering partners may still have spammers as customers. In the case of transit peering, transitive trust relationships occur, making the assumption even less realistic that no SPIT calls will originate in a peering provider’s network.

Trust in callers (or in their calls not being SPIT) does not play a role for our system. Instead, we implicitly establish trust in SSPs’ call tagging capabilities, represented by SPIT probabilities for tuples (Provider identity, algorithm, score assigned by algorithm) as trusted entities. Each SSP trusts a small set of other providers correctly report these SPIT probabilities. This small set is what we refer to as “trusted providers”. The procedure for selecting trusted providers is out of scope for this paper; section IV will show the impact of the number of trusted providers on the quality of SPIT assessments. A small number of trusted providers makes it easy to deal with untruthful reports, for example with legal means, based on contract relationships. This possibility makes it unattractive for any provider to give false feedback about SPIT likelihoods. Therefore, we think it is reasonable not to explicitly consider the possibility of an attack by a trusted provider. Throughout the rest of this document, will denote the set of a provider’s trusted providers. Optionally, we support one level of transitive trust:\footnote{We believe that some providers might allow transitive trust, but that due to the risk of trust network infiltration at most one level of transitive trust is reasonable.} Let \( U(x) \) be the set of all providers trusted by \( x \), either directly or transitively (i.e. including those trusted by \( x \)’s directly trusted providers).

\[
U(x) = \bigcup_{y \in T(x)} T(y) \cup T(x) \tag{1}
\]

\[C. \text{ Information Storage and Exchange}\]

Each SSP is assumed to have a relational database, in which the following attributes are stored based on its own experience as well as the input from its (transitively) trusted providers:

- **Target:** The provider that is rated.
- **Algorithm:** The algorithm used to compute the SPIT score.
- **Score:** The SPIT score given by Target, using Algorithm.
- **SPIT, No_SPIT, No_Feedback:** The number of calls for each of the categories.
- **FromDate, ToDate:** The timespan over which the information has been aggregated.
- **Source:** The SSP this information is from (either the local provider or a trusted provider).

At certain time intervals, each SSP sends contents of this database directly to all its trusted providers. Despite our design choice of aggregating reputation information on the receiver’s side, no SSP will be willing to dump its complete database and send it to other companies, since locally collected information is likely to be stored on a high level of detail. Instead, several filtering steps have to be performed, depicted in figure 1.

The first step is obfuscation. To preserve company secrets, the numbers of SPIT and other calls may be changed to just indicate an order of magnitude, while leaving the ratio of SPIT calls roughly intact. Next, all information already known to the recipient will be filtered out. The sender will then remove all data sets that do not originate from itself (or, if transitive trust is supported, from its own directly trusted providers). To avoid DoS attacks to the reputation system by a large number of provider identities and using each of them for just one call, the relevance filter removes all data sets for providers for which the number of legitimate calls is lower than a pre-defined threshold. Finally, all information that has not changed significantly since the last information exchange will not be sent. The recipient will assume this information to be unchanged; therefore, if information about a certain entity drops below the relevance threshold for the first time, an explicit notification is necessary. The recipient will perform a plausibility check first. In case of transitive trust, the recipient may receive information about a rated entity by more than one trusted provider. Then, only one of the sources is considered to be authoritative. Also, the recipient may decide aggregating the received information on a different timescale than the sender.

\[D. \text{ Evaluating SPIT Scores}\]

Obviously, the aim of our work is to improve the classification of incoming calls as SPIT (or legitimate) calls. An SSP using the system is not constrained to a specific computation of a call’s SPIT probability. Instead, it may use any information provided by the system in an arbitrary way, for example using sophisticated trust models or taking historic data into account. However, as our focus is on the overall system rather than on local optimizations, we restrict ourselves to a simple example of how to assess a call. At first, we assume that an outgoing call contains one tag of the form (algorithm used, score assigned by this algorithm).

1) **Tagging with a SPIT probability:** In the simplest case, an SSP receiving a call will assume the SPIT probability of this call to be equal to the observed percentage of SPIT calls received from the same provider and tagged with the same value—based on end users’ feedback. The receiving SSP takes into account its own observations and those of its trusted providers in the last time period. Using trusted providers’ data increases the sample size used for the estimation, leading to a reduced variance and thereby allowing an improved SPIT likelihood assessment.

Assume a set \( S \) of \( n \) providers and a set \( A \) of algorithms being used in the system. Assume further that provider \( r \in S \) (receiver) receives a call from provider \( s \in S \) (sender). \( r \) wants to assess the probability that the call is spit, \( P_{spit}(s, a_s, \mu_s(a_s)) \), based on \( s \)’s identity, the algorithm
$a_i \in A$ used by $s$, and the score $\mu_s(a_i)$ assigned by this algorithm. $\mu_s(a_i)$ can be a score with arbitrary semantics, while $P_{spit}(s, a_i, \mu_s(a_i))$ is a translation into an actual SPIT probability. If no tag is present, this can be considered as a special case with algorithm NULL and score NULL—our proposed algorithm loses information in this case, but it still works. Let $M_r(s, \mu_s(a_i))$ be the number of malicious (SPIT) calls with score $\mu_s(a_i)$ that $r$ received from $s$ in the previous time period. Analogously, $L_r(s, \mu_s(a_i))$ is the number of legitimate calls with that same score. Let $\tau (0 \leq \tau \leq 1)$ be a weight factor for providers that are only transitively trusted. With $\tau = 0$, no transitive trust is supported. Using its trusted providers, $r$ can compute $P_{spit}(s, a_i, \mu_s(a_i)) = \frac{1}{1 + \frac{\sum_{x \in T} L_x(s, \mu_s(a_i)) + \tau \sum_{x \in U} M_x(s, \mu_s(a_i))}{M_r(s, \mu_s(a_i)) + \sum_{x \in T} L_x(s, \mu_s(a_i)) + \tau \sum_{x \in U} M_x(s, \mu_s(a_i))}}$ (2) Note that this way of tagging does not simplify “whitewashing” by creating new identities, even if allowed by the underlying peering model. Only a sufficient number of legitimate calls (to trustworthy providers) decreases the SPIT percentage as reported by the system. Especially, if no legitimate calls have been received from a provider, $P_{spit}(s, a_i, \mu_s(a_i))$ is 1. Figure 2 visualises how a SIP-provider $B$ can use the reputation system to assess a SPIT-tag received in an incoming message from provider $A$: Upon receiving a tag $(A, a_i, \mu_A(a_i))$ (5), provider $B$ can query its local database, $RS_B$ (6), to receive the probability that this message is SPIT, $P_{spit}(A, a_i, \mu_A(a_i))$ (7). $RS_B$ can compute this probability using equation 2 and the information it received from its trusted providers ($C - F$). Based on this probability (most probably in combination with other SPIT-prevention modules $B$ uses locally), $B$ can decide on how to process the message further (8), e.g., forwarding it to the callee (9) if it is regarded unlikely to be SPIT.

2) Combining different tags: A provider has the choice to add more than one tag to outgoing invite messages. In this case, the receiving provider $r$ has to derive a combined score from $m$ different single scores $P_{spit}(s, a_i, \mu_s(a_i))$ ($i = \{1, \ldots, m\}$) with $m$ algorithms used by $s$. A similar situation arises when several SSPs are involved in the forwarding/redirecting of a SIP message. Each of them may add its own tags; the receiver can choose to consider any subset of tags. With no other information given, the combined rating could be computed as the average of all the provided tags’ ratings (whether different tags were added by one provider or by several in a chain). The use of more advanced algorithms to compute a combined score is possible, but out of this paper’s scope.

3) Embedding the system in a SPIT prevention framework: If applied as a stand-alone SPIT prevention system, the presented approach suffers from a bootstrapping problem: As long as no legitimate calls have been placed by a new provider, its calls will be considered as SPIT ($P_{spit}$ is 1 in equation 2). But while this is the case, they might be filtered before anyone has the chance of classifying them as legitimate. We therefore consider our approach to be part of a bigger SPIT prevention framework, as presented by Schlegel et al. [10]. Other modules present in such a framework can lead to the classification of a call as legitimate, even though the reputation system considers it to be SPIT with a high probability. Moreover, instead of blocking a call, it might be forwarded to a voice mailbox and classified as legitimate later on. As the voice telecommunication business is less dynamic than the provisioning of e-mail services, the addition of new, legitimate providers is likely to be a rare event, anyway. Using a reputation system even simplifies the bootstrapping process in comparison to a local whitelist, since positive experience from trusted providers can be used in addition to local experience. Thus, the overall detection quality—and even the number of false positives—are likely to be improved with our approach.

IV. ANALYSIS

The reputation system we propose in this paper has two advantages: it provides a semantic mapping of SPIT tags and it provides a way to better assess the quality of SPIT-detection algorithms. The first benefit is of qualitative nature. In this section, we evaluate the quantitative impact of the latter benefit through analytical calculations. For simplicity, we only demonstrate the improvement when our system is used with binary SPIT-algorithms, i.e., algorithms which tag a message with either 0 for non-spit or 1 for spit.4

A. Formal Model

We use the following formal model: The system consists of a set $S$ of $n$ providers, each having $u$ users. In each epoch $e_i$, $\omega$ calls originate at each provider $p_i \in S$, with destinations distributed uniformly among all providers in the system (including $p_i$). For each provider $p_i \in S$ we have $\alpha(p_i) \times \omega = \sigma(p_i)$ outgoing SPIT calls and $(1 - \alpha(p_i)) \times \omega = \nu(p_i)$ outgoing non-spit calls in each epoch $e_i$. In this evaluation section we assume that at each provider $p_i$ only one binary algorithm $a_i \in A$ is used, where $A$ is the set of all algorithms being applied in the network. Each algorithm

---

4Note that with more granularity in tags the system is even more beneficial because in that case the receiving SSP would have less data per tag in its own database for assessing tags. To this extent, the calculations in this section provide a lower bound on the benefit of our reputation system. On the other hand, a higher granularity requires more information to be exchanged and stored, so a reasonable tradeoff must be found.

5This is a simplification: In reality, a provider is going to receive a high number of calls from a small set of providers and a small number of calls from a large set of providers. Our reputation system is particularly useful for the latter calls, most notably the benefit for a receiving provider is larger for these calls compared to uniform distribution because the receiving provider gains relatively more information to judge these calls. 
For an algorithm \( a_i \) used by provider \( p_i \), we define the precision as follows: \( \text{Prec}(a_i) = P(\text{message is SPIT} | \text{message is tagged as SPIT by algorithm } a_i) \). We can calculate the precision of an algorithm \( a_i \) as follows, assuming that \( \omega, \alpha(p_i), \varphi(a_i), \) and \( \pi(a_i) \) are known [19]:

\[
\text{Prec}(a_i) = \frac{\sigma(p_i) \times (1 - \pi(a_i))}{\sigma(p_i) \times (1 - \pi(a_i)) + \nu(p_i) \times \varphi(a_i)} \tag{3}
\]

Accordingly, the recall of a message, i.e., \( P(\text{message is tagged as SPIT by algorithm } a_i | \text{message is SPIT}) \), can be calculated with the following equation [19]:

\[
\text{Rec}(a_i) = 1 - \pi(a_i) \tag{4}
\]

\( a_i \) used by provider \( p_i \) has a false positive rate \( \varphi(a_i) \) and a false negative rate \( \pi(a_i) \).

\( a_i \) with the following equation \( \{19\} \):

\[
\text{Rec}(a_i) = 1 - \pi(a_i) \tag{4}
\]

a) Algorithm-Precision/Recall without the proposed System: We now assume that an incoming SSP wants to assess the SPIT probability of an incoming message tagged using algorithm \( a_i \). This is equal to \( \text{Prec}(a_i) \) if the message is tagged as SPIT, and can be easily derived from \( \text{Rec}(a_i) \), if the message is tagged as Non-SPIT. We therefore calculate the variance of a receiving provider’s estimation of algorithm precision and recall. For simplicity, we assume three different classes of algorithms, according to their false positive rate and false negative rate (i.e., algorithms with good, medium, or bad SPIT-detection, and assuming a tradeoff between \( \varphi(a_i) \) and \( \pi(a_i) \)) and three different types of providers (i.e., providers from which small, medium, or large amounts of SPIT emerge):

- \( \forall a_i \in \{A,B,C\} \) with \( P(\text{class } a_i = x) \) or \( P_i(x) \) for short. Accordingly, we denote the probability of an arbitrary provider \( p_i \) belonging to type \( y \in \{A,B,C\} \) with \( P(\text{type } p_i = y) \) or \( P_i(y) \) for short. The average precision when considering all algorithms and messages is \( \text{Prec}_\text{avg} = P(\text{message is SPIT} | \text{message is tagged as SPIT by arbitrary algorithm}) \)

\[
\text{Prec}_\text{avg} = \frac{\sum_{x \in \{1,2,3\}} \sum_{y \in \{A,B,C\}} P_i(x) \times \text{Rec}(a_i) \times \text{Prec}(a_{y|x})}{\sum_{x \in \{1,2,3\}} \sum_{y \in \{A,B,C\}} P_i(x) \times \text{Rec}(a_i) \times \text{Prec}(a_{y|x})} \tag{5}
\]

\( a_{y|x} \) is an algorithm of type \( y \) and class \( x \) and \( \text{Prec}(a_{y|x}) \) gets computed according to Definition 3. Accordingly, the average recall can be calculated as follows:

\[
\text{Rec}_\text{avg} = \frac{\sum_{x \in \{1,2,3\}} P_i(x) \times \text{Rec}(a_i)}{\sum_{x \in \{1,2,3\}} P_i(x) \times \text{Rec}(a_i)} \tag{6}
\]

Note, however, that a provider \( r \in S \) (receiver) which receives a message being tagged as SPIT from a provider \( s \in S \) (sender) can only judge the precision/recall of the used algorithm based on the number of messages it received in the previous epoch from \( s \), which is \( \frac{u}{n} \) when calls are uniformly distributed among providers. We can consider the sampling of these received messages as a Bernoulli trials process. Thus, the variance of the receiver’s estimation of algorithm precision is, for an average message [20]:

\[
\text{Var}(\text{Prec}_\text{avg}, r) = \frac{\text{Prec}_\text{avg} \times (1 - \text{Prec}_\text{avg})}{\frac{n}{u} \times (1 + |T(r)|)} \tag{7}
\]

The variance of recall estimations can be obtained accordingly; we omit that equation due to space limitations.

b) Algorithm-Precision/Recall with the proposed System: When the receiving provider \( r \) uses the reputation system we propose, its estimation of the sending provider’s algorithm precision gets better. Depending on the number of trusted providers of \( r \), \( |T(r)| \), the variance of the precision and recall estimations for average messages, using the reputation system \( RS \) get computed as follows (because there are more Bernoulli samples in \( RS_r \)):

\[
\text{Var}(\text{Prec}_\text{avg}, RS_r) = \frac{\text{Prec}_\text{avg} \times (1 - \text{Prec}_\text{avg})}{\frac{n}{u} \times (1 + |T(r)|)} \tag{8}
\]

\[
\text{Var}(\text{Rec}_\text{avg}, RS_r) = \frac{\text{Rec}_\text{avg} \times (1 - \text{Rec}_\text{avg})}{\frac{n}{u} \times (1 + |T(r)|)} \tag{9}
\]

B. Evaluation

Our rationale is that algorithms with different detection quality exist in the system and that most providers have only very few malicious users while a small percentage of providers has many spitters. Thus, we assume the following distribution among algorithm-classes and provider-types: \( P_i(1) = 0.5, P_i(2) = 0.3, P_i(3) = 0.2, P_i(A) = 0.7, P_i(B) = 0.25, P_i(C) = 0.05 \). Figure 3 and 4 show the variance of precision (and recall, respectively) estimations for different numbers of trusted providers \( |T(r)| \) (where \(|T(r)| = 0 \) implies that the receiving provider only trusts itself) and different systems with \( n \) providers and \( u \) users at each provider (displayed as \( [n; u] \)). The figures illustrate that our system considerably improves SPIT-algorithm estimation for a receiving provider. Further,
the results show that the reputation system provides the biggest advantage if many SSPs with a comparatively small number of users exist. Intuitively, in that case a receiving SSP has less own past experience with a caller’s SSP, and can gain most from our system.

### C. Using Transitive Trust among Providers

To analyse the system when one level of transitive trust is used, we calculate \(|U(r)|\) (assuming \(|T(r)|\) to be equal for all SSPs, and assuming trusted providers to be drawn according to a uniform distribution): We start with \(r\)'s directly trusted SSPs, of which there are \(\zeta_0 = |T(r)|\). In each step, we then add the trusted providers of one more of these directly trusted SSPs (for \(i > 0\): \(\zeta_i = \zeta_{i-1} + |T(r)| - \frac{|T(r)\times n-i}{n-1}\)), where the minuend is the expected value of a hypergeometric distribution, taking into account the chance of drawing already (transitively) trusted providers. \(|U(r)|\) is equal to \(\zeta(|T(r)|)\).

Depending on the weight \(\tau\) the receiving provider wants to put on the transitively trusted providers, the variance of precision (recall) estimations can be obtained as

\[
\text{Var}(\text{Rec}_{avg}) = \frac{1}{1 + |T(r)| + |U(r)| + |T(r)|\times \tau} \times \text{Var}(\text{Rec}_{avg}) \times \left(\frac{|T(r)|\times n-i}{n-1}\right) \times \tau.
\]

Tables I and II show the results of this calculation for different \(|T(r)|\) and \(\tau\). It can be observed that one level of transitive trust significantly improves algorithm assessment for the receiving provider (compare with figures 3 and 4). For instance, when \(|T(r)| = 2\), \text{Var}(\text{Rec}_{avg}) can be reduced from 0.21 (see figure 3) up to 0.10.

### V. Conclusion

We have presented a system that uses provider-level reputation to improve SPIT prevention. As a unique feature of this system, we provide an incentive for providers to tag outgoing calls according to their SPIT likelihood. The combination of an outgoing provider’s identity with a tag assigned to the message is considered as an entity to be rated by the reputation system. This way, attack detection can be improved by making use of information only available to the caller’s provider. At the same time, the reputation system can be used to assess the meaning of tags assigned by the caller’s provider, e.g., by translating them into a SPIT probability. We have shown analytically that SPIT detection can be improved significantly by taking advantage of other providers’ experience.

Despite these advantages, the presented approach can only be part of a SPIT prevention solution. Future research will focus on how to integrate our reputation system into an existing VoIP security framework, and will also investigate whether results can be improved by evaluating providers’ past tagging quality and SPIT call numbers using more advanced classification models.

### References


