HAL
archives-ouvertes.fr

An Analysis for Anonymity and Unlinkability for a VoIP Conversation

Ge Zhang

▶ To cite this version:

Ge Zhang. An Analysis for Anonymity and Unlinkability for a VoIP Conversation. 5th IFIP WG 9.2, 9.6/11.4, 11.6, 11.7/PrimeLife International Summer School(PRIMELIFE), Sep 2009, Nice, France. pp.198-212, 10.1007/978-3-642-14282-6_16 . hal-01061069

HAL Id: hal-01061069
https://hal.inria.fr/hal-01061069

Submitted on 5 Sep 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution| 4.0 International License

An Analysis for Anonymity and Unlinkability for A VoIP conversation

Ge Zhang

Karlstad University, Karlstad, Sweden
ge.zhang@kau.se

Abstract. With the growth of its popularity, VoIP is increasingly popular nowadays. Similarly to other Internet applications, VoIP users may desire to be unlinkable with their participated VoIP session records for privacy issues. In this paper, we explore the Items of Interests (IOIs)[1] from anonymisation aspects based on a simplified VoIP model and analyse the potential links between them. We address possible methods to break the links. Finally, we also discuss requirements for a VoIP anonymisation Service (VAS) in terms of functionality, performance and usability. Based on this, we discuss the fundamental design requirements for a VAS which we intend to subsequently implement.

1 Introduction

Voice over IP (VoIP) is a method which enables users to build voice conversation with partners at a distance by transferring encoded voice data over packet-switched networks (e.g., the Internet). Like other applications in the Internet, VoIP users may search for privacy protection, such as anonymity. For example, a caller may prefer to withhold his/her real identity to others (even a callee) for various reasons. However, to the best of our knowledge, most existing VoIP anonymity solutions are based on a single Trusted Third Party (TTP) and no detailed analysis of VoIP anonymity in terms of unlinkability has been done. Thus, we are motivated to explore this area further. Similar to the terminologies in [1], we define **caller anonymity**, which means to a potential VoIP caller, each requested VoIP conversation record is unlinkable; **callee anonymity**, which means to a potential VoIP callee, each accepted VoIP conversation record is unlinkable; **relationship anonymity**, which means for a given VoIP conversation record, its caller and callee are unlinkable. Caller/callee anonymity is a stronger property than relationship anonymity: an attacker has to break both caller anonymity and callee anonymity in order to break relationship anonymity. Thus, as long as caller/callee anonymity is achieved, relationship anonymity is achieved as well. *In this research, we focus on the caller/callee anonymity for standardized VoIP services.* First, we enumerate the Items of Interest (IOIs) based on a simplified VoIP model. Second, we depict the potential relationship among these IOIs and

[1] The IOIs in this paper refer to a piece of information which an attacker is interested to know

explain how a user can be tracked by observing the relationship among the IOIs. Based on this, we suggest methods to break the relationships to make the actual user unlinkable to a VoIP session. Finally, we propose some further requirements on a VoIP anonymisation Service (VAS) which we will investigate in our future work.

The rest of this paper is organized as follows. Section 2 introduces standardized VoIP protocols. Some previous works on VoIP anonymity and their limitations are presented in Section 3. Section 4 shows our analysis of VoIP anonymity in terms of unlinkability. Section 5 proposes requirements of a VAS. We discuss our considerations on a VAS design in Section 6. Section 7 provides conclusions of this paper.

2 SIP-based VoIP

Current standards for VoIP protocols and services are standardized by the Internet Engineering Task Force (IETF). Among these protocols, three of them are essential: the Session Initiation Protocol (SIP) [2], the Session Description Protocol (SDP) [3] and the Realtime Transport Protocol (RTP) [4]. SIP is designed as a signaling protocol aiming at establishing, modifying or terminating a session between users. A SIP server[2] provides SIP clients with multiple services such as locating users, relaying SIP messages, etc. While SDP is designed for the purposes of session announcement, session invitation. A SDP message is generally contained in a SIP message as a message payload. After negotiating by using SIP and SDP messages, users can build a media session with each other. The session can comply with RTP protocol, which provides end-to-end network transport functions suitable for bidirectional transmitting encoded voice packets over network services. SIP and SDP communications are generally classified as *signaling layer* while RTP communications are characterized as *media layer*.

2.1 Signaling Layer

A SIP user needs to login to the domain of the SIP server for requesting the services. In this way, a SIP client sends the SIP server a REGISTER message including a SIP Uniform Resource Identifier (URI) and the networking location [3] of the user. The format of SIP URI is similar to an Email address, consisting of a pair of user name and domain name, (e.g., "sip:ge.zhang@kau.se"), which represents a user with username "ge.zhang" at "kau.se" domain. After receiving the REGISTER message, the SIP server will keep the mapping between the URI and its networking location. Therefore, the SIP server is able to locate and forward messages to this user later. The procedure is as follows:

[2] We assume the SIP server includes all necessary service components (e.g., SIP proxy server, SIP registrar, SIP redirect server) in this paper.

[3] In most cases, the location information is denoted by the IP address and port number of a client's User Agent (UA).

1. $user \longrightarrow server : REGISTER < URI_{user}, Location_{user}, ... >$
2. $server \longrightarrow user : 200OK$

After successful registering, the SIP users are able to call or to be called. Here, we describe how a VoIP session can be built by using SIP. We assume that there are three entities involved in this model: the caller, the callee and the server. The caller would like to build a conversation with the callee, so the caller knows the URI_{callee}, but does not know the $location_{callee}$. Therefore, the caller needs the server's help to setup a signaling communication to the callee. In this way, the caller first sends the server an INVITE request, including $URI_{caller}, URI_{callee}, Location_{caller}$, etc. The server should know $Location_{callee}$ if the callee has been already registered on the domain. In this way, the server can forward the INVITE message to the callee. The callee then responses a 200OK message to accept the calling request. The 200OK message, including the location of the callee, will be forwarded to the caller. Finally, the caller sends the callee an ACK message to acknowledge the calling request. Then, the conversation should be established without the participation of the server. The procedure is as follows:

1. $caller \longrightarrow server : INVITE < URI_{caller}, URI_{callee}, Location_{caller}, ... >$
2. $server \longrightarrow callee : INVITE < URI_{caller}, URI_{callee}, Location_{caller}, ... >$
3. $callee \longrightarrow server : 200OK < URI_{caller}, URI_{callee}, Location_{callee}, ... >$
4. $server \longrightarrow caller : 200OK < URI_{caller}, URI_{callee}, Location_{callee}, ... >$
5. $caller \longrightarrow callee : ACK$
6. $caller \longleftrightarrow callee$: Conversation on media layer

2.2 Media layer

An RTP session consists of two kinds of data streams: one for the actual encoded voice data stream and another for control information, named as RTP Control Protocol (RTCP) [5]. Three features of RTP sessions are especially important for our research:

– Transmitted voice data is encoded and decoded using a special purpose speech codec algorithm (e.g., G.711 [6] and Speex [7]) negotiated in the signaling level. The codec takes the voice from users as input, which is typically sampled at either 8k samples or 16k samples per second (Hz). As a performance requirement, the inter-arrival packet time of voice stream is generally fixedly selected between 10 and 50 ms, with 20 ms being the common case. Thus, given a 8 kHz voice source, we have 160 samples per packet with 20 ms packets interval. Moreover, the size of each voice packet depends on the encoding bit rate of adapted codec. Two types of encoding bit rate can be distinguished: **Fixed Bit Rate (FBR)** and **Variable Bit Rate (VBR)**. With FBR (e.g., G.711), end points produce voice packets always with the same size. On the other hand, VBR (e.g., Speex) means that the encoding bit rate varies according to the type of voice. Therefore, end points produce voice packets with different size.

- RTP allows discontinuous transmission (**silence suppression**) [8], which is a capability of endpoints to stop sending RTP packets during silent periods of its owner. In this circumstance, additional resources (e.g., bandwidth) can be saved. However, whether to use silence suppression is usually a configuration option for users.
- Each endpoint periodically sends control packets by using RTCP to the other side [4]. The control packets contain information about the received and transmitted data rates, delay jitter and packet losses. A RTCP communication generally uses a different communication channel from voice data communication.
- The Secure Realtime Transport Protocol (SRTP) [9] specifies a new RTP profile to provide confidentiality, integrity protection and data origin authentication to the RTP and RTCP traffic. SRTP requires a key exchange mechanism to generate session keys for encrypting and decrypting the voice data traffic. The key exchange mechanisms are classified into signaling level (e.g., MIKEY [10]) and media level (e.g., ZRTP [11]), depending on whether the exchanging is taken place in signaling traffic or the media traffic.

3 Related Work

RFC3323 [12] endeavored to design a mechanism which enables SIP users to launch anonymous calls. To achieve this goal, some identity information in SIP messages (e.g., user's URI, IP address of User Agent (UA), etc) should be concealed from other subjects. The author thus proposed two kinds of privacy-enhanced mechanisms: *user-provided privacy* and *network-provided privacy*. User-provided privacy mechanism is designed for a requirement of low-level anonymity. With this mechanism, optional personal information is removed from SIP messages (e.g., a SIP message can optionally contain a URL pointing to an online photo of the caller. As an optional information, this kind of URL should be automatically stripped by a user-provided privacy). The actual VoIP call is not impacted without these optional information. However, the effect of this mechanism is rather limited: users' URI and the IP addresses of their equipments still appear in SIP messages. Without these information, the SIP servers do not know where the responses of these messages should be forwarded. Thus, RFC 3323 suggested the network-provided privacy mechanism, in which a privacy server, working as a trusted third party, constantly converts the user's URI in a SIP message to a randomized pseudonym. A privacy server also should keep the mapping state of the user's URI and the pseudonym for the routing purpose. Based on RFC 3323 [12], Charles Shen, et al., [13] proposed a more comprehensive analysis on identity leaking of SIP messages. They further represented an architecture with a privacy server, which was implemented according to the specifications of RFC 3323. However, their solutions heavily rely on a single Trusted Third Party (TTP). Nevertheless, a single TTP-based anonymisation service is insufficient to provide a high-level protection: It can be broken as long as the TTP is manipulated or compromised by attackers.

4 The analysis of caller/callee anonymity

In this section, we list potential *IOIs* with their relationships in the VoIP context based on the VoIP model in Section 2. Then, we analyse how a VoIP user can be traced by exploiting the relationships.

4.1 Item of Interest (IOIs)

Much information can be revealed during a VoIP communication. A more comprehensive analysis of personal information leaking in SIP and SDP messages has been discussed in [13]. However, personal information is supposed to be minimised if the user searches for privacy protection, which means a privacy-aware VoIP user does not provide any personal information unless they have to. Taking this condition into account, we list several potential IOIs based on the simplified VoIP model proposed in Section 2 as follows:

1. **VoIP user:** A VoIP user is referred to the actual person who utilizes VoIP services.
2. **SIP Service Provider (SP):** A SIP SP provides SIP services for VoIP users in a specific SIP domain.
3. **SIP URI:** A SIP URI is a VoIP user's identifier on the signaling level.
4. **Networking location:** As voice and signaling packets are transmitted over packet-switched networks, a networking location, especially, an IP address is used for locating a user's equipment. It is a user's identifier on the networking level.
5. **VoIP session:** A VoIP session refers a conversation of two users on media layer.

According to [1], "Linkability of two or more IOIs from an attacker's perspective means that within the system, the attacker can suffiently distinguish whether these IOIs are related or not." Our analysis is based on the simplified VoIP context as described in Section 2. There are three entities involved in the VoIP context, with a caller, a callee, and a SIP SP. We assume a user (either the caller or the callee) would like to withhold "who called whom" for privacy reasons. However, a potential attacker targets at observing "who called whom" from the information in the conversation. We assume that *the SIP SP and the user on the other side of the communication are potential attackers.* From an attacker's view, a VoIP user can be traced according to the links depicted in Figure 1. The links are shaped in this way: Given a VoIP session, one IOI might be fully or partly deduced from another. The representations of the numbered links are as follows:

Link 1 (VoIP session → SIP URI): In order to establish a VoIP session, users on both sides should first exchange SIP messages with each other for signaling establishment. Thus, the users' SIP URIs contained in SIP messages are revealed and related to a certain VoIP session. And it is known by all participates (the caller, the callee, and the SIP SP).

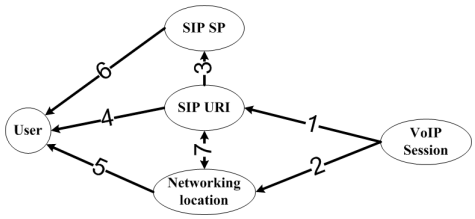


Fig. 1. Linkable IOIs in a VoIP context

Link 2 (VoIP session → Networking location): A SIP SP knows their users' networking locations in order to relay and to forward SIP messages for them. Furthermore, both the caller and the callee should know each other's networking location to build a VoIP session. In this way, users' networking locations are related to a certain VoIP session, which is known by all participates (the caller, the callee, and the SIP SP).

Link 3 (SIP URI → SIP SP): As introduced in Section 2, a SIP URI contains a domain name of the SIP SP which the user registers to. Thus, it is fairly clear that a user's SIP SP is indicated by the domain name. For instance, given a SIP URI "sip:ge.zhang@kau.se", we say that the user registers to the SIP SP with domain name "kau.se".

Link 4 (SIP URI → User): A SIP URI can be used to trace its owner in the following ways:

- By username: Besides a domain name, a SIP URI also contains a username of a user. Some users take real names as their usernames in favor of others to remember. For example, a SIP URI, "sip:ge.zhang@kau.se", exposes the real name of its owner.
- Linked with calling records: A user can select a pseudonym URI in which the user's real name does not appear. For example, "sip:batman@iptel.org". However, a pseudonym URI does not mean that its owner is untraceable. If a user participated in a set of calls with a single pseudonym URI, these calling records might be useful information to trace the actual user.

Link 5 (Networking Location → User): A Networking location (IP address) can be used to track its owner in the following ways:

- WHOIS lookup: WHOIS [14] provides publicly available information that allows one to query a remote WHOIS database for registration information of a domain name. Generally, a WHOIS record contains a full name, address, telephone number and email address of the Internet Service Provider (ISP). A WHOIS search accepts IP address as an input for querying. In this way, it forms a relationship between the owner of an IP address and its ISP.
- Geographical location: There are a lot of online services [15] which provides mappings between an IP address and a geographical location of its Internet Service Provider (ISP) (including country and city). It can be effective to locate a VoIP user from his/her IP address.

- Linked with other Internet applications: As a user can access a variety of services (e.g., web, email, etc) in the Internet besides VoIP, a user generally may reuse one IP address for different applications. In this case, different applications can be linked by a single IP address, which makes the user easier to be tracked.

Link 6 (SIP SP → User): A SIP SP can be used to trace a user in the following ways:

- Relationship: Since the information of most SP is available in the Internet, it may reveal the relationship between a SP and its users. For example, for a given SP "kau.se", an attacker can find out that kau.se is a domain name of Karlstad University in Sweden. Thus, the attacker can further guess that the users of this SP are either students or faculty members at this university.
- Limited number of users: Some SP may contain only a small number of users, which decreases the user's anonymity set. In this way, users of this SIP SP are easier to trace.

Link 7 (SIP URI ↔ Networking location): Users have to register in their SP domain for services by providing their SIP URI and networking locations. Thus, a SP can deduce a user's networking location from SIP URI and vice versa.

4.2 An analysis for anonymity and unlinkability

In this section, we endeavor to find a mechanism which enables users to achieve caller/callee anonymity, which means to a potential SIP user (whatever caller or callee), each VoIP session is unlinkable. As shown in Figure 1, a user is traceable from the links. Thus, we discuss the feasibility of breaking these links.

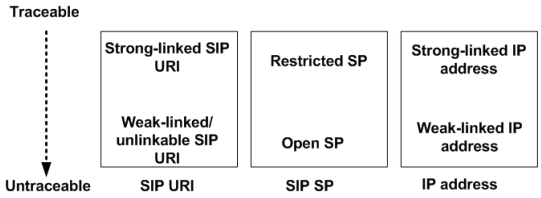


Fig. 2. A brief taxonomy of SIP URI, SP and IP address according to the linkability to their owner

Breaking link 1: To break link 1, a SIP URI should not appear in SIP signaling for a VoIP setup. This obviously contradicts the specification of SIP protocol in RFC 3261 [2]. Therefore, breaking link 1 is unrealistic.

Breaking link 2: To break link 2, networking location (IP address) should not be revealed to build a VoIP session, which defies both SIP protocol[2] and RTP protocol[4]. Thus, breaking link 2 is also not viable.

Breaking link 3: To break link 3 means that the actual domain name of a SIP SP should not appear in a SIP URI. However, a URI is malformed without a domain name, which might cause unexpected behavior of SIP infrastructures. As a result, breaking link 3 is unrealistic as well.

Breaking link 4: We first roughly separate SIP URIs into two category according to the linkability between a SIP URI and its owners.

- **Strong-linked SIP URI**: A SIP URI is defined as a strong-linked SIP URI if personal information exposed from this SIP URI is sufficient to trace its owner. For example, it can be a SIP URI containing the real name of its owner.
- **Weak-linked/unlinkable SIP URI**: A SIP URI is defined as a weak-linked/unlinkable SIP URI if personal information exposed from this SIP URI is insufficient to trace its owner. For instance, It can be a one-time URI only used once per call (unlinkable), or a shared URI which can be potentially used by a number of people (weak-linked).

To break link 4, users should employ weak-linked SIP URI.

Breaking link 5: We first roughly separate IP addresses into two category according to the linkability between an IP address and its owners.

- **Strong-linked IP address**: An IP address is defined as a strong-linked IP address if personal information exposed from this IP address is sufficient to track its owner. For example, an IP address which is used at a specific company, or an IP address which is used at a user's home.
- **Weak-linked IP address**: An IP address is defined as a weak-linked IP address if personal information exposed from this IP address is insufficient to trace its owner. For example, a user made a VoIP call at an Internet cafe in a foreign country. In this case, the IP address used in the internet cafe offers the user a higher anonymity than the one used at home or office. However, different to SIP URI, one-time IP addresses are difficult to provide due to the constraint of current IPv4 address space.

To break link 5, users should employ weak-linked IP address.

Breaking link 6: A classification of SIP SPs is discussed below:

- **Restricted SP**: A restricted SIP SP aims to provide services only to the users in a specific group or an organization. For example, a SIP domain "kau.se" only provides SIP services to the faculty members or the students at Karlstad University. The relationship between a SIP SP and its user is then revealed, which is useful to find out a specific VoIP user.
- **Open SP**: There are a lot of open SIP SPs, of which the services are not limited to a specific group, but available to all Internet users. The relationship between an open SP and its users is simply as VoIP service-client. An open SP provides better anonymity than a restricted SP.

To break link 6, using an open SP is recommended instead of a restricted SP.

Breaking link 7: Users should register their SIP URI and networking locations to their SP according to SIP specification. Thus, breaking link 7 is unrealistic as well.

So far, we have discussed methods to break the links in Figure 1. It is difficult to break link 1, 2, 3 and 7 unless we modify VoIP protocols, but it is possible to break link 4, 5, 6 by using various SIP URIs and IP addresses. A summary of the taxonomy of SIP URI, SP and IP address is illustrated in Figure 2. Weak-linked URIs, weak-linked IP addresses and open SPs are recommended for anonymity. Therefore, for example, a user can make a call anonymously in this way: He/She can setup a call with an one-time URI and an open SP by using a computer at an Internet cafe. However, it is not a scalable solution and it is inconvenience for users to do this in reality. Therefore, we are motivated to design and implement a VoIP anonymisation Service (VAS) to help users to achieve anonymity in an easier way.

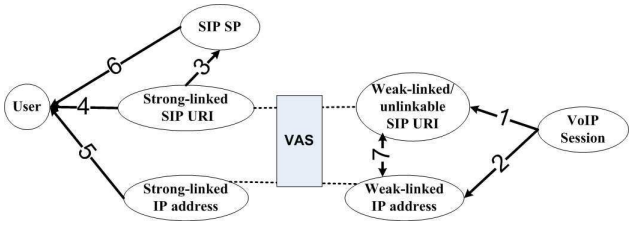


Fig. 3. A perspective VoIP anonymisation service (VAS) provides a higher anonymity to users by mapping SIP URI and IP address

Shown in Figure 3, a VAS should constantly map both the strong-linked SIP URI and IP address to weak-linked or unlinkable ones. Some further requirements of this VAS is discussed in the next section.

5 Requirements for a VAS

Several requirements for the VAS must be taken into account to be suitable for a VoIP environment. These are describes below in the contexts of basic requirements, performance, usability and resistance to traffic analysis.

5.1 Basic requirements

We consider the following basic requirements of functionalities of a VAS.

- anonymisation service: The VAS provides anonymisation services by mapping a user's strong-linked SIP URI and IP address to the weak-linked or unlinkable ones.
- Compliant to VoIP protocols: A VAS should be compliant to existing VoIP protocols standardized in RFC documents. Although SIP protocol supports extension to some degree, the extension should be minimised for scalability. Furthermore, a VAS should be designed to understand the grammar defined in VoIP protocols since it needs to map SIP URIs and relay VoIP sessions.

– UDP support: The VAS must support UDP communication. TCP, being designed to provide a reliable end-to-end communication with a "flow control" method, has been employed for many Internet services (e.g., web services and file delivery). However, in return, the "flow control" method consumes additional bandwidth. On the other hand, UDP, without a "flow control" method, is more efficient than TCP. Similar to other stream media applications, VoIP session does not need a reliable communication since a small amount of packet loss actually does not prevent users from understanding the whole conversation. Therefore, RTP protocol was designed on the top of UDP to achieve a better performance instead of reliability. Furthermore, SIP protocol can work over both TCP as well as UDP. Thus, to be compatible with existing VoIP protocols, our VAS will be designed to be accessed over UDP.
– No single Trusted Third Party (TTP): Most privacy problems can be easily tackled by introducing a single TTP. However, in reality, a single TTP may not be relied on. Our solution should avoid single point of failure.

5.2 Requirements on performance

For voice communication over packet-switched networks, three issues generally affect the quality of service including network delay, delay jitter and packet loss. These three issues are frequently taken as criteria to evaluate performance of communication services.

– Network delay: It refers to the time interval elapsed from the moment one user sends a voice packet until the user at another side receives the packet. It is mainly caused by transmission, propagation and queuing of packets. The network delay affects the voice conversation when the delay reaches a certain threshold. According to [16], the delay of a voice communication will not affect users as long as it is less then 150 ms; When the delay is between 150 to 400 ms, the quality of conversation is still acceptable but users will notice a slight hesitation in their partner's response. While the delay is above 400 ms, the performance is unacceptable for voice communication since the users cannot follow the conversation.
– Delay jitter: It is caused by the network congestion and improper routing during the transmission of voice packets. As a result, the packets arrive the receiver side at an uneven rate, which can lead to short-term audio gaps if the delay jitter is too large. According to [16], the performance is unacceptable if the value is above 75 ms for most codecs used.
– Packet loss: Some voice packets may be dropped or discarded during the transmission. As said above, VoIP conversation is able to endure packet loss, however, too much packet loss can lead to an incomplete conversation. Packet loss may occur due to many reasons (e.g., traffic congestion at a router in the middle). The impact introduced by packet loss varies generally depending on the codec design.

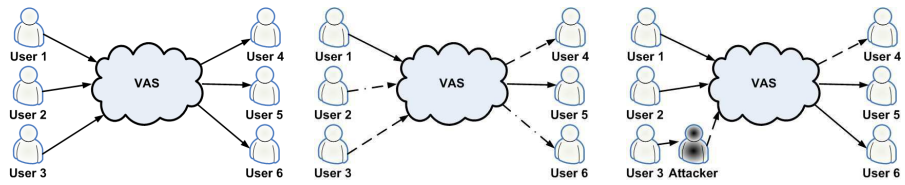
Employing a VoIP VAS may introduce negative impact on the performance of packets transferring since signaling and voice packets have to traverse over additional networking nodes as "stepping stones". It probably leads to more network delay, delay jitter and packet loss rate. Thus, to make a VAS useable, the values of network delay, delay jitter and packet loss rate introduced must be kept in an acceptable level as described above.

5.3 Requirements on usability

A VoIP user may be confused when operating a VAS if its user interface is too complicated. Also, a VoIP user may be too inexperienced to use a VAS. Thus, we are motivated to implement a user-friendly interface which supports following functions. The user interface should be designed easy-to-use. Some predefined privacy settings should be built-in with the interface.

5.4 Requirements on resistance to traffic analysis

Different with Section 4, here we extend the threat model considering more practical requirements: The attackers not only can be the SP and the communication partners, but also can be intermediaries in the network. In this way, we need to take traffic analysis attacks into account. Traffic analysis attack aims to correlate the flows entering and leaving a VAS by observing their characteristics (e.g., size and inter-arrival time of packets) of the flows. A flow entering the networking and a flow leaving the network can be paired if they have similar characteristics. For example, an attacker cannot correlate the flows in Figure 4(a) as all flows look the same. It is hard to say whom user1 communicates with (could be user4, user5, or user6). However, each flow has its own characteristics in Figure 4(b). In this case, attackers can easily correlate the flows by their characteristics (user1 $\leftrightarrow$ user5, user2 $\leftrightarrow$ user4, user3 $\leftrightarrow$ user6). Instead of passively observing, Attackers can also modify a flow to insert more characteristics before it enters the network (active attack), as illustrated in Figure 4(c). Our VAS is designed to prevent both the passive and the active traffic analysis attacks.



(a) No characteristic can be distinguished for flows

(b) Each traffic flow has its own characteristic

(c) An attacker injects characteristics into a flow

Fig. 4. Traffic analysis attacks on a VAS

6 Work in progress

We plan to construct a VAS by using anonymous overlay network, which is a virtual network built on the top of other network layers to hide a user's real identifier. The anonymity service is then provided by the nodes in the anonymous network, instead of a single TTP. Some anonymous overlay networks are already available in the Internet to provides anonymity services for different applications, which can be mainly divided into two categories: One for high-latency applications (e.g., mail) and another for low-latency applications (e.g., web surfing). In the later case, both the Tor [17] and the AN.ON [18] have been operated in the Internet for several years. The Tor network forwards users' traffic through several routers with multi-layer encryption, with each router decrypting one layer of the encryption. The end-to-end path is constructed by several circuits between end-points and Tor routers in a telescoping fashion. In this way, each Tor router only knows the previous and the next router in the network, but it has no idea of the whole end-to-end path. Generally, Tor routers are dynamically selected from the network prior to the communications or during the communications. On the other hand, AN.ON employs *cascades*, which consist of predefined mixing routers. Thus, AN.ON users select cascades instead of routers.

R. Wendolsky, et al [19] provide an empirical study of the performance comparison of Tor and AN.ON. Their results show that Tor is subject to unpredictable performance (with average end-to-end delay from 2000 ms to 7500 ms), while AN.ON can provide more consistent performance in general (with average end-to-end delay from 1000 to 1500 ms). This is mainly caused by their topology. Recently, [20], [21] and [22] addresses the alternatives to enhance the performance of Tor network by router selecting algorithms based on some parameters (e.g., on bandwidth, latency, etc). While optimising performance, the paths are not randomly selected anymore, which reduces anonymity. The authors also proposed their metrics for tradeoff performance and anonymity. However, without a proper measurement mechanism on performance [23], these router selecting algorithms cannot be employed by Tor so far.

Usually, anonymous overlay networks employ different techniques (e.g., packets-padding, dummy traffic and packets-delaying, etc) to eliminate (or hide) the characteristics of flows to mitigate traffic analysis attacks. In return, these techniques usually lead to a worse networking performance, which means they cannot be easily employed for VoIP services. As an alternative, we take the features of VoIP into account and discuss how the features can be exploited to prevent traffic analysis attacks.

- As said, VoIP endpoints will not send packets during silence period if silence suppression is enabled. Thus, it introduces additional timing characteristics to VoIP flows. The work [24] shows that it is easy to pair VoIP flows by these characteristics. Without silence suppression, VoIP endpoints can send voice packet with a fixed rate, which means that less characteristics of inter-arrival time between packets can be used for traffic analysis. *Thus, silence suppression must be disabled for VAS.*

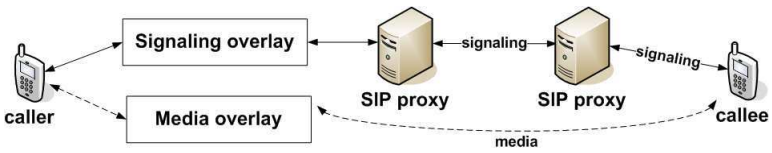


Fig. 5. Two anonymous overlay networks should be used, one for signaling and another for media

- VoIP endpoints will generate voice packets with different sizes for a given conversation if a VBR is specified instead of a FBR. This also introduces a VoIP flow with more characteristics which makes a VAS more vulnerable to traffic analysis attacks. Besides traffic analysis, [25] and [26] demonstrates that packets size varying can reveal the conversation content between two VoIP users even the traffic is encrypted. *Therefore, FBR is highly recommended for VAS instead of VBR.*
- Similarly to the design in [27], we consider a defensive dropping mechanism to defend active traffic analysis attacks. As mentioned, the voice packets for silence period are meaningless for the actual conversation, but they must be sent to reduce flow characteristics. As an alternative, these silent "packets" can also be randomly dropped at the Mixer routers to obscure the characteristics. In this way, the traffic analysis is difficult even if the attackers actively introduce timing characteristics in the flow (e.g., [28]). The "silence packets" can be marked by the original endpoints, with indicating that which router can drop which packets.
- The end-to-end delay for voice conversation should be less than 450 ms. We can also consider a dropping mechanism based on time stamp. We say that a voice packet transmitted for 500 ms is less important than one transmitted for 100 ms. Therefore, we consider a QoS-aware scheme to minimize the impact of traffic congestion. For example, we set a timeout, saying 450 ms. The packets which have already spent more than 450 ms in the transmission should be dropped in case of traffic congestion.
- There are many different features between signaling traffic and media traffic. First, they have different requirements on performance: signaling traffic can suffer more delay (several seconds) but less packet loss, while with media traffic it is just the opposite. Moreover, media traffic can be composed by packets with the same size and inter-arrival time. However, signaling traffic cannot achieve this. Taking these difference in mind, we plan to employ two overlay network for each respectively (shown in Figure 5).
- As introduced in Section 2, endpoints periodically send RTCP control packets to the other side in default. With different packet sizes, RTCP streams are vulnerable to traffic analysis attacks. However, RTCP stream is designed to be optional and independent from RTP voice data stream. In this way, end-points should disable RTCP when they access a VAS. Moreover, the key exchange mechanism should be taken place in signaling traffic instead of media traffic to minimize the characteristic of media traffic.

7 Conclusion and future work

This study provided an investigation of VoIP anonymity in terms of unlinkability. We demonstrated that for a VoIP user, the privacy requirements on the signaling level and the session level should not be considered separately. We also proposed requirements towards a VAS including functionality, performance and usability. However, this work did not show any concrete solution of the VoIP VAS, as this is work in progress. In our future work, we are going to design the VAS in more detail. We consider to construct a VAS and do some experiments on the performance. For example, we are curious to find out how much performance can be enhanced by the packet dropping method addressed in Section 6 and in which condition our VAS solution can provide services with end-to-end delay less than 450 ms in a large-scale networking environment (e.g., the Internet).

Acknowledgments. The author would like to thank Stefan Köpsell, Stefan Berthold, Sebastian Clauss and Professor Fischer-Hübner for their comments and suggestions.

References

1. A. Pfitzmann and M. Hansen. Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management - a consolidated proposal for terminology. Technical report, February 2008.
2. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler. SIP: Session Initiation Protocol, 2002. RFC 3261.
3. M. Handley and V. Jacobson. SDP: Session Description Protocol, 1998. RFC 2327.
4. H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson. RTP: A Transport Protocol for Real-Time Applications, 2003. RFC 3550.
5. I. Johansson and M. Westerlund. Support for Reduced-Size Real-Time Transport Control Protocol (RTCP):Opportunities and Consequences, 2009. RFC 5506.
6. G.711. http://www.itu.int/rec/T-REC-G.711/e, visited at 21th-Oct-2009.
7. Spexx. http://www.speex.org/, visited at 21th-Oct-2009.
8. R. Zopf. Real-time Transport Protocol (RTP) Payload for Comfort Noise (CN), 2002. RFC 3389.
9. M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman. The Secure Real-time Transport Protocol (SRTP), 2004. RFC 3711.
10. J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman. MIKEY: Multimedia Internet KEYing, 2004. RFC 3830.
11. P. Zimmermann, Ed. A. Johnston, and J. Callas. ZRTP: Media Path Key Agreement for Secure RTP draft-zimmermann-avt-zrtp-15, 2009. Internet-Draft.
12. J. Peterson. A Privacy Mechanism for the Session Initiation Protocol (SIP), 2002. RFC 3323.
13. C. Shen and H. Schulzrinne. A voip privacy mechanism and its application in voip peering for voice service provider topology and identity hiding, 2008.
14. K. Harrenstien, M. Stahl, and E. Feinler. NICNAME/WHOIS, 1985. RFC 954.
15. Geobytes. http://www.geobytes.com/IpLocator.htm?getlocation, visited at 10th-May-2009.

16. S. Karapantazis and F. Pavlidou. Voip: A comprehensive survey on a promising technology. *Comput. Netw.*, 53(12):2050–2090, 2009.
17. R. Dingledine, N. Mathewson, and P. Syverson. Tor: the second-generation onion router. In *SSYM'04: Proceedings of the 13th conference on USENIX Security Symposium*, pages 21–21, Berkeley, CA, USA, 2004. USENIX Association.
18. O. Berthold, H. Federrath, and S. Köpsell. Web mixes: a system for anonymous and unobservable internet access. In *International workshop on Designing privacy enhancing technologies*, pages 115–129, New York, NY, USA, 2001. Springer-Verlag New York, Inc.
19. D. Herrmann R. Wendolsky and H. Federrath. Performance comparison of low-latency anonymisation services from a user perspective. In *PET'07: Privacy Enhancing Technologies*, pages 233–253, Berlin / Heidelberg, 2007. Springer LNCS.
20. R. Snader and N. Borisov. A tune-up for tor: Improving security and performance in the tor network. In *NDSS'08: Proceedings of the Network and Distributed System Security Symposium, NDSS 2008*. The Internet Society, 2008.
21. M. Sherr, B. T. Loo, and M. Blaze. Towards application-aware anonymous routing. In *HOTSEC'07: Proceedings of the 2nd USENIX workshop on Hot topics in security*, pages 1–5, Berkeley, CA, USA, 2007. USENIX Association.
22. S. J. Murdoch and R. N. Watson. Metrics for security and performance in low-latency anonymity systems. In *PETS '08: Proceedings of the 8th international symposium on Privacy Enhancing Technologies*, pages 115–132, Berlin, Heidelberg, 2008. Springer-Verlag.
23. R. Dingledine and S. J. Murdoch. Performance improvements on tor or, why tor is slow and what we're going to do about it, 2009.
24. M. Vlachos, A. Anagnostopoulos, O. Verscheure, and P. S. Yu. Online pairing of voip conversations. *The VLDB Journal*, 18(1):77–98, 2009.
25. C. V. Wright, L. Ballard, F. Monrose, and G. M. Masson. Language identification of encrypted voip traffic: Alejandra y roberto or alice and bob? In *SS'07: Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, pages 1–12, Berkeley, CA, USA, 2007. USENIX Association.
26. C. V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson. Spot me if you can: Uncovering spoken phrases in encrypted voip conversations. In *SP '08: Proceedings of the 2008 IEEE Symposium on Security and Privacy*, pages 35–49, Washington, DC, USA, 2008. IEEE Computer Society.
27. C. Wang B. N. Levine, M. K. Reiter and M. Wright. Timing attacks in low-latency mix systems (extended abstract). In *FC '04: Proceedings of the 8th international conference on Financial Cryptography*, pages 251–265, Berlin, Heidelberg, 2004. Springer-Verlag.
28. X. Wang, S. Chen, and S. Jajodia. Tracking anonymous peer-to-peer voip calls on the internet. In *CCS '05: Proceedings of the 12th ACM conference on Computer and communications security*, pages 81–91, New York, NY, USA, 2005. ACM.