A VoIP Privacy Mechanism and its Application in VoIP Peering for Voice Service Provider Topology and Identity Hiding

Charles Shen and Henning Schulzrinne
Department of Computer Science
Columbia University
{charles, hgs}@cs.columbia.edu

October 3, 2006

Abstract

Voice Service Providers (VSPs) participating in VoIP peering frequently want to withhold their identity and related privacy-sensitive information from other parties during the VoIP communication. A number of existing documents on VoIP privacy exist, but most of them focus on end user privacy. By summarizing and extending existing work, we present a unified privacy mechanism for both VoIP users and service providers. We also show a case study on how VSPs can use this mechanism for identity and topology hiding in VoIP peering.

1   Introduction

Privacy mechanisms for SIP-based VoIP can be found in a number of existing documents, including current RFCs (RFC 3323 [1], RFC 3325 [2], RFC 3261 [3], RFC 3711 [4]), an obsolete RFC (RFC 2543 [5]), and an expired internet draft (Byerly [6]). One motivation of our work is to summarize useful pieces of related contents in these documents and present them as a unified privacy mechanism for VoIP.

The other motivation of our work comes from requirements raised by Voice Service Providers (VSPs) that participate in VoIP peering. VoIP peering is the interconnection of VSPs at the session layer for exchanging SIP-based signaling among each other. The peering VSPs together form a large end-to-end SIP-enabled signaling network. This network allows individual peering VSPs to greatly extend the service diversity and coverage they can provide to their customers, bypassing the Public Switched Telephone Network (PSTN). The ability to avoid routing traffic through PSTN also cuts significant service costs. Despite the advantages of peering, many VSPs concern about their identity and network topology information being exposed to other service providers in the peering network. VSP identity exposure potentially enables the VSP's competitors to learn about who its customers are and then target specific marketing efforts to win them over. VSP topology information exposure could reveal the VSP's internal network structure, network size and raise business concerns as well as security threats. Therefore, VSPs frequently desire to protect their privacy-sensitive information in VoIP peering. Existing privacy work on VoIP, however, mostly focuses on privacy requirements for end users. For example, RFC 3323 defines privacy as "the withholding of the identity of a person (and related personal information) from one or more parties in an exchange of communications, specifically

a SIP dialog. These parties potentially include the intended destination(s) of messages and/or any intermediaries handling these messages." Although most user privacy mechanisms also provide certain privacy protection for service providers, directly applying user privacy mechanisms does not solve the service provider privacy problem completely. Moreover, protecting service provider privacy does not always entail protecting user privacy at the same time.

In this report, we broaden the RFC 3323 definition of privacy to cover service providers. Specifically, privacy in this document is defined as "the withholding of the identity of a person (and related personal information) and/or a service provider (and related service provider information, such as its network topology) from one or more parties in an exchange of communications, specifically a SIP-based VoIP session." We extend existing work and present a mechanism for both user and service provider privacy. We start with identifying message fields in key VoIP protocols that potentially reveal user or service provider information. Then we describe the privacy service functions that can be performed at the user side and at the network intermediaries. We show how user level and service provider level privacy requirements can be supported by those privacy functions. Using the service provider privacy component of the privacy mechanism, we gave a case study on how the VSPs can achieve identity and topology hiding in a generic VoIP peering architecture.

The rest of this report is organized as follows: Section 2 describes a privacy mechanism for SIP-based VoIP; Section 3 introduces VoIP peering architecture and gives a case study of achieving VSP privacy in VoIP peering. Section 4 concludes the report.

2   A Privacy Mechanism for SIP-based VoIP

2.1   Privacy-Sensitive Message Fields in SIP, SDP, RTP and RTCP

The SIP header fields that may reveal user or service provider information are summarized in Table 1. The first column gives the header type with an example header format, some of which are drawn from Section 20 of RFC 3261. The second column indicates the privacy characteristics of the corresponding field. Since our privacy definition includes privacy-sensitive information for both user and service provider, this column contains a "u" for user-related information fields, and a "p" for service provider-related information fields.

Common user-related information includes the user name, email, SIP address, phone, location, and information about the subject of the session. Note that the host name or IP address where the session media starts is also considered user related information.

Service provider-related information is usually presented in the form of a domain name or IP address. This can be seen in Alert-Info, Call-ID, Call-Info, Contact, Error-Info, From, In-Reply-To, Proxy-Authenticate, Record-Route, Reply-To, Route, To, Via, Warning, and WWW-Authenticate header fields. Most of the header fields that reveal service provider information contain the user name or user display name and therefore also disclose user information. These header fields include Contact, From, In-Reply-To, Reply-To, and To. In fact, all these header fields may contain SIP URI, which is commonly formed by a user part and a service provider part. SIP also allows the use of other URI formats. For example, tel URI [7] and Globally Routable User Agent URIs (GRUU) [8]. Some of the tel URI parameters, such as Phone-Context, may expose identity about the service provider. The GRUU may contain a SIP address of record, and therefore may expose both user and service provider information.

The privacy characteristics of some of the header fields is less straightforward and depends highly on how they are used. For example, the Organization header field contains the "name of the organization to which the SIP element issuing the request or response belongs [3]", it may or may not be related to the user or the service provider; the Subject header field "provides a summary or indicates

2

Header field example	privacy
Alert-Info: <http://vsp.example.com/sounds/moo.wav>	p
Authorization: Digest username="Alice", realm="VIP@example.com", nonce="84a4cc6f3082121f32b42a2187831a9e", response="7587245234b3434cc3412213e5f113a5432"	u
Call-ID:f81d4fae-7dec-11d0-a765-00a0c91e6bf6@192.0.2.4	up
Call-Info: <http://vsp.example.com/alice/photo.jpg>;purpose=icon, <http://vsp.example.com/alice/>;purpose=info	p p
Contact: "Alice" <sips:alice@vsp.example.com>;expires=60	up
Error-Info: <sip:not-in-service-recording@vsp.example.com>	p
From: "Bob" <sip:bob@vspexample.com>;tag=hyh8	up
In-Reply-To: 10230@vsp.example.com, 44150@vsp.example.com	up
Organization: example Inc.	up
Proxy-Authenticate: Digest realm="VIP@example.com", domain="sip:vsp.example.com", qop="auth", nonce="f84f1cec41e6cbe5aea9c8e88d359", opaque="", stale=FALSE, algorithm=MD5	p
Record-Route: <sip:vsp1.example.com;lr>, <sip:vsp2.example.com;lr>	p
Reply-To: Bob <sip:bob@vsp.example.com>	up
Route: <sip:vsp1.example.com;lr>, <sip:vsp2.example.com;lr>	p
Server: HomeServer v2	p
Subject: Tech Support	u
To: sip:+12125551212@vsp.example.com	up
User-Agent: Softphone Beta1.5	p
Via: SIP/2.0/UDP vsp.example.com:5060; branch=z9hG4bK87asdks7	p
Warning: 307 example.com "Session parameter 'foo' not understood"	p
WWW-Authenticate: Digest realm="VIP@example.com", domain="sip:vsp.example.com", qop="auth", nonce="f84f1cec41e6cbe5aea9c8e88d359", opaque="", stale=FALSE, algorithm=MD5	p

Table 1: SIP header fields that contain privacy-sensitive user and service provider information

the nature of the call [3]" and will likely affect user privacy; the User-Agent and Server header fields contain information about the SIP user agent client and user agent server. In some cases, these information can be linked to the service provider, especially when the service provider is also providing the SIP user equipment.

SIP uses SDP to describe media streams in VoIP sessions. The SDP fields that may reveal user or service provider information are listed in Table 2. SIP session media is commonly carried in RTP [9]. RTP is usually used together with its control protocol RTCP. One type of RTCP packets called Source

Description (SDES) packets contain privacy-sensitive information as listed in Table 3. Most of the fields that reveal service provider information contain a domain name or IP address, similar to the case in SIP. The RTCP SDES tool field is analogous to the SIP User-Agent and Server header fields. Whether it reveals information about the service provider depends on actual usage situation.

Field name	Example	Privacy
Origin	o=charles 2890844526 2890842807 IN IP4 128.59.66.4	up
Email	e=charles@cs.columbia.edu	up
Phone	p=+12125551234	u
Session-name	s=VoIP seminar	u
Information	i=A seminar on VoIP	u
URI	u=http://www.cs.columbia.edu/ charles/sdp.ps	up
Connection	c=IN IP4 128.59.66.4	up

Table 2: SDP message fields related to user and service provider privacy

Field name	Example	Privacy
Cname	doe@sleepy.example.com	up
Name	Charles	u
Loc	Morningside, Manhattan	u
Email	charles@example.com	up
Tool	videotool 1.2	p

Table 3: RTCP SDES packet fields related to user and service provider privacy

There are two important points we want to emphasize about the above privacy-sensitive fields. First, the "u" or "p" privacy characteristics of a particular field is assigned based on a common usage pattern. In some cases, there are different ways to fill in the field that can prevent it from exposing privacy-sensitive information, at the same time without affecting its validity. Indeed this property is used in the privacy mechanism discussed in the later part of this document. Second, the service provider information fields may actually correspond to different types of service providers. For example, the IP address in the SDP Origin field may disclose the user's Internet Access Provider (IAP) identity; the domain name part of the SIP URI in the SIP Contact header may disclose the user's VSP; the domain name in the RTCP SDES packet's Email field may disclose the user's Email service provider. These different types of service providers could all be the same one or be completely different, or be of any other combination. Therefore, it is impossible to precisely correlate these fields to one particular type of service provider without knowing the actual service provider relationship.

2.2 User Side Privacy Service Functions

This section discusses possible privacy functions the user side can perform. For SIP and SDP, the user side functions are carried out by a SIP user agent. For RTP, the user side functions are carried out by an RTP client. For convenience, we copied part of the table on "Summary of (SIP) header fields" from RFC 3261 in Table 4. We only list those header fields that could reveal privacy-sensitive information as indicated in Table 1. The notations used in Table 4 are also copied as below:

Header field	where	proxy	ACK	BYE	CAN	INV	OPT	REG
Alert-Info	R	ar	-	-	-	o	-	-
Alert-Info	180	ar	-	-	-	o	-	-
Authorization	R		o	o	o	o	o	o
Call-ID	c	r	m	m	m	m	m	m
Call-Info		ar	-	-	-	o	o	o
Contact	R		o	-	-	m	o	o
Contact	1xx		-	-	-	o	-	-
Contact	2xx		-	-	-	m	o	o
Contact	3xx	d	-	o	-	o	o	o
Contact	485		-	o	-	o	o	o
Error-Info	300-699	a	-	o	o	o	o	o
From	c	r	m	m	m	m	m	m
In-Reply-To	R		-	-	-	o	-	-
Organization		ar	-	-	-	o	o	o
Proxy-Authenticate	407	ar	-	m	-	m	m	m
Proxy-Authenticate	401	ar	-	o	o	o	o	o
Proxy-Authorization	R	dr	o	o	-	o	o	o
Record-Route	R	ar	o	o	o	o	o	-
Record-Route	2xx, 18x	mr	-	o	o	o	o	-
Reply-To			-	-	-	o	-	-
Route	R	adr	c	c	c	c	c	c
Server	r		-	o	o	o	o	o
Subject	R		-	-	-	o	-	-
To	c(1)	r	m	m	m	m	m	m
User-Agent			o	o	o	o	o	o
Via	R	amr	m	m	m	m	m	m
Via	rc	dr	m	m	m	m	m	m
Warning		r	-	o	o	o	o	o
WWW-Authenticate	401	ar	-	m	-	m	m	m
WWW-Authenticate	407	ar	-	o	-	o	o	o

Table 4: Summary of SIP header fields

The "where" column describes the request and response types in which the header field can be used. Values in this column are:

R: header field may only appear in requests;

r: header field may only appear in responses;

2xx, 4xx, etc.: A numerical value or range indicates response codes with which the header field can be used;

c: header field is copied from the request to the response.

An empty entry in the "where" column indicates that the header field may be present in all requests and responses.

The "proxy" column describes the operations a proxy may perform on a header field:

a: A proxy can add or concatenate the header field if not present.

m: A proxy can modify an existing header field value.

d: A proxy can delete a header field value.

r: A proxy must be able to read the header field, and thus this header field cannot be encrypted.

The next six columns relate to the presence of a header field in a method:

c: Conditional; requirements on the header field depend on the context of the message.

m: The header field is mandatory.

m*: The header field SHOULD be sent, but clients/servers need to be prepared to receive messages without that header field.

o: The header field is optional.

t: The header field SHOULD be sent, but clients/servers need to be prepared to receive messages without that header field. If a stream-based protocol (such as TCP) is used as a transport, then the header field MUST be sent.

*: The header field is required if the message body is not empty. See Sections 20.14, 20.15 and 7.4 for details.

-: The header field is not applicable.

"Optional" means that an element MAY include the header field in a request or response, and a user agent MAY ignore the header field if present in the request or response (The exception to this rule is the Require header field discussed in 20.32). A "mandatory" header field MUST be present in a request, and MUST be understood by the user agent server receiving the request. A mandatory response header field MUST be present in the response, and the header field MUST be understood by the user agent client processing the response. "Not applicable" means that the header field MUST NOT be present in a request. If one is placed in a request by mistake, it MUST be ignored by the user agent server receiving the request. Similarly, a header field labeled "not applicable" for a response means that the user agent server MUST NOT place the header field in the response, and the user agent client MUST ignore the header field in the response.

The user side can use two common methods to assist in privacy protection. The first method is removal and anonymization of privacy-sensitive information in the generated messages. For SIP message headers, the removal applies to SIP header fields that are optional in the signaling process. From Table 4, we see that the SIP Call-Info, In-Reply-To, Organization, Reply-To, Server, Subject, User-Agent, Alert-Info, Error-Info, and Warning header fields belong to this category. Exactly which of these fields should be removed depends on whether user privacy or service provider privacy needs to be protected. If user privacy is desired, the header fields among them with privacy characteristics "u" should be removed. If service provider privacy is desired, the header fields among them with privacy characteristics "p" should be removed. If both user and service provider privacy are desired, all these header fields should be removed. For SDP description part (Table 2), the Email, Phone, Information, and URI fields may be omitted. For RTCP SDES packet fields (Table 3), the Name, Location, Email, and Tool fields may be omitted.

Anonymization at the user side applies to some of the mandatory fields that could reveal privacy-sensitive information. The exact way to anonymize these fields depends on whether user privacy or

service provider privacy needs to be protected. If user privacy is to be protected, the host name or IP address in SIP Call-ID and In-Reply-To header fields, as well as the host name in RTCP SDES Cname field, should be replaced by a suitably long random value; the user display name in SIP Contact, From, and To header fields should be replaced by *anonymous*; the SIP URIs that are not used for routing further messages within the same SIP dialog should be anonymized to `sip:anonymous@anonymous.invalid`. As an example, the anonymized form of a From header may become `From: "Anonymous" <sip:anonymous@anonymous.invalid>`.

If service provider privacy is to be protected, anoymization should be applied specifically to the service provider related part of the privacy-sensitive information. Therefore, the host name or IP address in SIP Call-ID and In-Reply-To header fields, as well as the host name in RTCP SDES Cname field, should still be replaced by a suitably long random value; the user display name in SIP Contact, From, and To header fields do not need to be worried about; the SIP URIs that are not used for routing further messages within the same SIP dialog should be partially anonymized. Taking the From header as an example again, its anonymized form for hiding service provider information may be `From: "Alice" <sip:alice@anonymous.invalid>`. It can be seen that only service provider information is concealed; the original user information is kept intact. One important note about the service provider privacy case is that, as we mentioned in Section 2.1, the fields we identified here may belong to different types of service providers. The privacy concern may be only for a subset of these service providers. For example, we may be interested in VSP identity hiding, not ISP identity hiding. Unless we have enough information to exactly locate the fields that are only related to the type of service provider we are interested in, we will have to act on all service provider privacy-sensitive fields shown in Section 2.1.

Note that authentication-related SIP headers need to be treated specially. RFC 3323 discusses these headers as follows: "Note that authentication mechanisms, including the Digest authentication method described in the SIP specification, are outside the scope of the privacy considerations in this document. Revealing identity through authentication is highly selective, and may not result in the compromise of any private information. Obviously, users that do not wish to reveal their identity to servers that issue authentication challenges MAY elect not to respond to such challenges."

The second method at the user side to protect privacy is encryption. For SIP messages (including SDP descriptions), RFC 3261 allows the user agent to protect SIP message privacy using S/MIME [10]. The whole SIP message can be encrypted using the tunneled "message/sip" MIME type, or a subset of the SIP message can be encrypted using the "message/sipfrag" MIME type. This end-to-end encryption mechanism is often used by the user to convey information to its destination user, without disclosing it to network intermediaries. It is suitable for certain header fields with end-to-end semantic, including Alert-Info, Authentication-Info, Error-Info, In-Reply-To, Organization, Reply-To, Server, Subject, User-Agent, and Warning. In some cases, the user also provides an encrypted version for those header fields that already have a clear text version. This is usually because the clear text header field may be modified in the network, but the origin user wants the destination user to receive the unchanged header value. For example, sometimes the SIP From header field may be anonymized in the network for privacy reasons. But the origin user wants the destination user to see the original From header value so it can properly display "caller-ID". This can be solved by inserting an encrypted version of the From header when the message is generated.

The encryption method can also be applied to RTP and RTCP. It is defined in RFC 3711 as part of the secure RTP mechanism.

2.3 Network Intermediary Privacy Service Functions

2.3.1 Network Intermediary-specific Privacy Service Functions

In both SIP signaling and RTP media, there are privacy-sensitive fields that cannot simply be concealed at the user side without affecting normal operation. For example, the SIP Contact, Via, Route, and Record-Route header fields contain URIs used for routing signaling messages within the current SIP dialog, they must be visible to the signaling routing element. The session IP address in SDP Origin and Connection fields also have to be valid. Finally, although not the focus of our document, some lower level information such as the IP source addresses in RTP and RTCP could reveal identity of the traffic source as well. These information usually can not be manipulated by the user side.

Solving the privacy problem for these fields requires the involvement of network intermediaries. Existing documents suggest the following three methods that the network intermediaries can use for the SIP Via, Route and Record-Route headers. These methods are also applicable to the Contact header field.

Stripping: RFC 3323 introduces the "stripping" method, which removes the corresponding header fields and replaces them with those created by the privacy service. For example, in Via stripping, a list of Via header fields can be replaced by a single Via header field corresponding to the network entity performing the privacy service.

Encryption: RFC 2543 and Byerly [6] propose the "encryption" method, which asks the privacy service to use a secret key to encrypt the corresponding header fields, with a timestamp and an appropriate checksum. Any network signaling entities beyond the one performing the privacy service will see the encrypted version of these fields. Only the original privacy service entity can decrypt these fields when the message containing these fields comes back.

Caching: RFC 2543 also mentions a "caching" method, which falls somewhere in between the "stripping" and "encryption" methods. The privacy service keeps a cache of the corresponding header fields and replaces them in the actual message with indexes into the cache. On the reverse path, the privacy service takes the header fields from the cache rather than from the message. It is suggested that the "encryption" method is favored over the "caching" method because of cache reuse concerns.

Two important conditions need to be satisfied in applying any of the above methods, namely, "recoverable" and "routable". "Recoverable" means the privacy service must make sure that it can restore the values for any of the header fields it has manipulated when further requests or responses within the same SIP dialog arrive. "Routable" means that any manipulation of those header fields by a network privacy service needs to make sure further messages within the same SIP dialog can be routed back to the same privacy service for header value recovery. Manipulating header fields in the network and subsequently restoring them necessarily require additional state information to be stored locally. Storage of excessive state information may have significant impact on the scalability of the entity providing privacy service. Among the three methods, the "encryption" method imposes the least burden on state management. However, encryption operation may require more processing power and incur longer processing latency.

It should be noted that the "encryption" and "caching" processing methods are not yet standardized. These methods were originally proposed in RFC 2543 and were then removed in the updated version of RFC 2543, or RFC 3261. We suggest that at least the "encryption" method be allowed as part of the SIP privacy framework. In fact, the main counterargument mentioned in RFC 3261 for the "encryption" method is "serious trust issue". But the trust concern can be alleviated in specific

application environments. VoIP peering is one such example where privacy service can be provided by the VoIP peering provider, and VSPs are assumed to trust their peering provider.

In addition to performing privacy service functions on SIP Via, Route and Record-Route headers, the network intermediary privacy service should also make sure that they do not add message fields containing privacy-sensitive information to the messages. Depending on whether user privacy or service provider privacy is concerned, these header fields may include Alert-Info, Call-Info, Error-Info, Server and Organization.

Network intermediary privacy services for RTP media usually involve a middlebox acting as media relay. It divides the media session into two segments so the original media sender and receiver will not see the opposite side directly. The IP addresses in the SDP Origin field and Connection fields will need to be modified accordingly.

A common type of network intermediaries that can implement the privacy service is SIP proxy. However, some operations require more functionality than what a SIP proxy can provide. According to Table 4, a SIP proxy is not allowed to modify or delete Record-Route and Via header fields in SIP requests. Therefore, when these operations are involved, the entity performing network intermediary-specific privacy service functions needs to act as a transparent Back-to-Back User Agent (B2BUA). B2BUA also has to be used when an RTP media relay is needed because in that case the SDP description of the SIP message body must be modified. A SIP proxy is not allowed to modify that part of the SIP message.

2.3.2 User Side Privacy Service Functions Performed at the Network Intermediary

If the user side is not able to perform its own privacy service functions, the network privacy service may need to act on the user's behalf. This applies to the removal and anonymization operations discussed in Section 2.2. Note that the exact operation and the affected fields depend on whether user privacy or service provider privacy needs to be protected.

The network intermediary performing SIP privacy service functions on behalf of the user can be a SIP proxy. But in some cases, it will need to act as a B2BUA. One such case is when any dialog matching field needs to be modified. As mentioned in RFC 3323, modification of the Call-ID header field is such an example. Modification of the From header field may also cause problem for older SIP implementations. Both Call-ID and From are header fields that need to be anonymized for privacy. Another case when B2BUA may be needed is when the Alert-Info, Error-Info, and Warning header fields have to be removed. According to Table 4, these fields cannot be deleted by a SIP proxy.

2.3.3 Indicating Different Network Privacy Service Levels

In general, both the user and the network intermediary may initiate privacy service functions. However, not all user clients and network entities are capable of privacy service functions. There ought to be a mechanism to convey the functions that need to be performed by the network privacy services. RFC 3323 defines a SIP Privacy header for that purpose. Five possible values for the Privacy header field are specified in RFC 3323. Each of those values is associated with a degree of privacy service as summarized below. *User* privacy assumes that the SIP user agent is not capable of privacy service functions, so it requires the network intermediary to perform SIP user side privacy functions protecting user privacy-sensitive information as we discussed in Section 2.3.2 and Section 2.2. *Header* privacy requires the network privacy service to perform SIP header related privacy functions that can only be done at network intermediaries as specified in Section 2.3.1. *Session* privacy requires the network intermediary to perform RTP media privacy service functions discussed in Section 2.3.1. *None* privacy

prohibits any privacy function to be performed. *Critical* privacy indicates that the request should be rejected if the corresponding privacy level cannot be accommodated. An additional value for the Privacy header is defined in RFC 3325, which is called *id* privacy. It means the SIP P-Asserted-Identity header should be removed before the message is transfered outside the trusted domain.

We extend the privacy definition of RFC 3323 to cover not only user information, but also service provider information. Consequently, we define a new *Service Provider* value for the SIP Privacy header to reflect that change. When this privacy value is specified, the network privacy service is required to perform SIP user side privacy service functions protecting service provider information as we discussed in Section 2.3.2 and Section 2.2.

There are also privacy service deployment considerations in delivering service provider privacy, especially service provider identity hiding. Usually even if a privacy service is deployed at the border of the network, the identity of the privacy service itself is still exposed to outside the network for routing purposes. Therefore, if a service provider wants to hide its identity, it needs to make sure that the identity of the network privacy service is distinct from its own identity. A common practice would be to use a third party network privacy service. It is also likely that the service provider's privacy service and the third party privacy service are used together. There are at least two benefits for this hybrid scheme. First, this allows the service provider to expose as little information to the outside as possible. The service provider can manage its topology hiding by its own privacy service and use the third party privacy service only for identity hiding. Second, the service provider's privacy service helps to offload part of the third party privacy service's state management burden. This allows the third party privacy service provider to scale better.

The same SIP Privacy header can be used to request privacy service from both the service provider and the third party privacy service. We noted that a specific rule in RFC 3323 states the following: "a Privacy header may include each legitimate privacy level value zero or one time. When the privacy functions corresponding to a requested privacy level is performed, the corresponding privacy level value is removed from the Privacy header. When the last privacy level value (excluding *critical*) is removed, the entire Privacy header should be removed." In the case where the service provider's own privacy service and the third party privacy service co-exist, the service provider's privacy service should not consider the privacy functions to be finished after its own processing, instead it should preserve the corresponding Privacy header values and pass them to the third party privacy service, where further processing will be done and the Privacy header value will be removed accordingly.

An alternative to requesting privacy service through the Privacy header is to make advanced arrangement with the privacy service provider.

2.4 Achieving High Level Privacy Requirements

The user side and network privacy service functions discussed in Section 2.2 and Section 2.3 form the building blocks to satisfy high level privacy requirements. With a privacy definition covering both the user and the service provider, we identify three common cases of high level privacy requirements and discuss how each of them can be satisfied. Variants of these requirements can be similarly accomplished based on solutions to these cases.

In the first case, a user wants to conceal user privacy-sensitive information both from network intermediaries and from the destination user. This can be achieved by applying the user side removal and anonymization functions for user privacy-sensitive information. If user side privacy service is not available, the network can provide the same functions with the SIP *user* privacy value in the Privacy header field. These operations provide reasonably good user privacy that is sufficient in most cases, although they do not guarantee that all user information is concealed. For example, these functions do not include anonymizing the SIP URI in the SIP Contact header, which is needed for routing

signaling messages. If the user wants to ensure strict privacy including those items, it will need to also request the SIP *Header* privacy from the network privacy service. Furthermore, if the user side is not able to perform user related privacy functions for RTP/RTCP, it will need to request the SIP *Session* privacy from the network to accomplish that.

In the second case, a user wants to conceal user privacy-sensitive information from network intermediaries involved in the communication, but not from its destination user. This case requires the techniques similar to those in the first case above. In addition to that, the user side SIP S/MIME and secure RTP/RTCP privacy function should be applied.

In the third case, a service provider wants to conceal service provider privacy-sensitive information from other service providers involved in the communication. To achieve this, the user side removal and anonymization service for service provider information should be applied. If user side privacy service is not available, the same service should be performed by the network with the SIP *service provider* privacy level. The network should also apply SIP *header* privacy. In addition, the identity of the network privacy service should be distinct from the identity of the service provider. Finally, the network may need to apply SIP *session* privacy in case any RTP/RTCP fields disclose information about the type of the service provider to be protected. Note that if there is no specific user level privacy concerns, the user may apply S/MIME to its SIP messages and allow the destination user to get all user information, without affecting the privacy concerns at the service provider level.

3 Providing Privacy for Service Providers in VoIP Peering

3.1 VoIP Peering Network Architecture

A VoIP peering partner network or VSP network usually consists of a common set of logical functions including Location Function (LF), Policy Function (PF), Signaling Function (SF) and Media Function (MF) [12]. The location function discovers the servers and hosts to be contacted through the help of location services such as DNS or ENUM. The policy function performs authentication and exchange policy parameters used by the signaling function. The signaling function performs the actual routing of SIP signaling messages. The media function may be optional because VoIP signaling and media do not necessarily follow the same path. When the media function is present, it is responsible for operations such as media transcoding and media security enforcement.

In a VoIP peering architecture, a VSP has two deployment options based on how the signaling and media relationship is managed, the composed and decomposed model. In the composed model, signaling and media follow the same path. In other words, the SF and MF are implemented in the same peering logical element. The communication between SF and MF in this model is straightforward. The problem with this model is that the combined SF and MF element creates a bottleneck and single point of failure. The decomposed model has distinct signaling and media paths and is more scalable than the composed model. The decomposed model can be further classified into quasi-decomposed and fully-decomposed models. In the quasi-decomposed model, the SF and MF are implemented in separate peering logical elements but usually both of them belong to the same service provider and there is a vertical control interface between the SF and MF. The communication between SF and MF in this model is more complicated than that in the composed model. In the fully-decomposed model, signaling and media are largely independent. There is no direct control between the SF and MF. In fact, the MF may not even need to exist. This model is the most scalable one, but the lack of control between SF and MF may not be desired when operational concerns such as billing and accounting are important. The composed and the two decomposed models of VSP as well as Voice Service Customer

(VSC) connecting to it the are illustrated in Figure 1.

(a) Composed model   (b) Quasi-decomposed model   (c) Fully-decomposed model
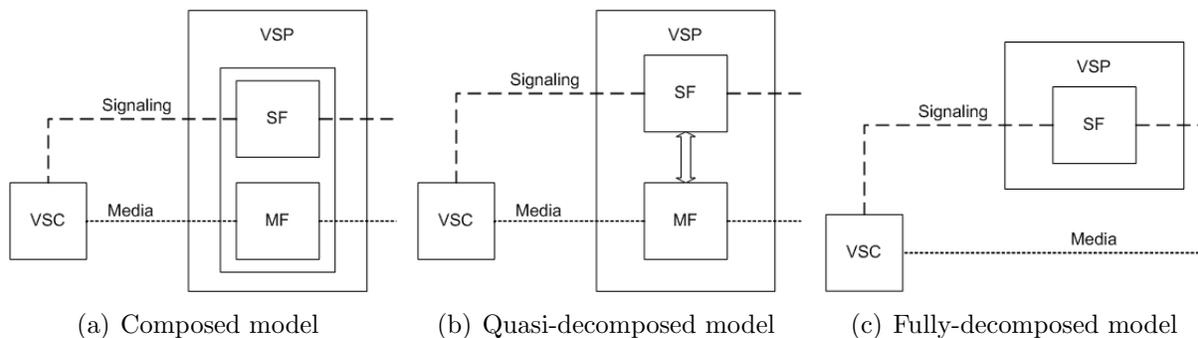
Figure 1: Different VSP deployment models

Since we are looking at the service provider privacy issue and customers may be associated with more than one service provider, the way customers connect to their service providers is another dimension we need to consider in the VoIP peering architecture. A VSC may be an enterprise network or an end-user. A VSC is usually associated with at least an Internet Access Provider (IAP) and obtains from this IAP data services or voice services or both. Figure 2 shows three common methods that a VSC use to connect to its ISPs. In Figure 2(a), the VSC uses a single IAP for both data service and voice service. An example of such type of service providers could be a cable company. In Figure 2(b) the VSC uses two different service providers for data and voice, but the connection to the VSP is through the IAP. Some residential end users may fall into this category. In Figure 2(c) the VSC uses two different service providers for data and voice, and the VSC has a separate connection to each of them. This may be the case for an enterprise VSC.

(a) Collocated VSP and IAP   (b) Separate VSP and IAP with a single connection   (c) Separate VSP and IAP with separate connections
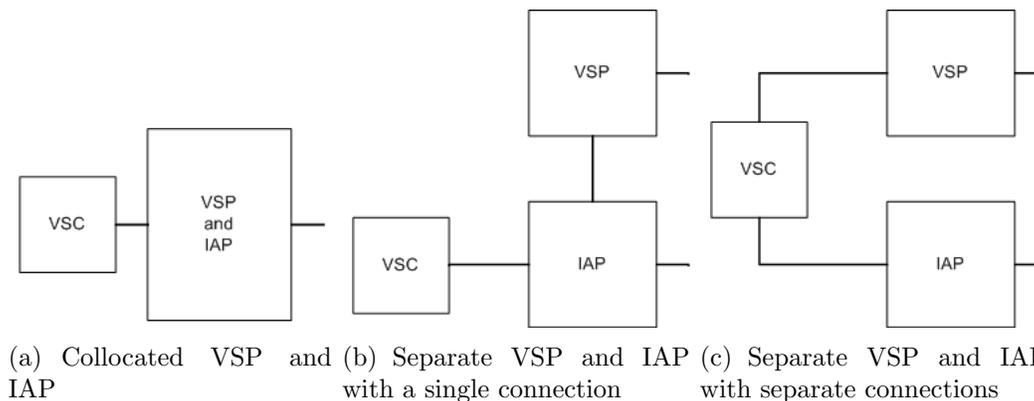
Figure 2: Different VSC and VSP connection methods

Another dimension of the VoIP peering architecture is whether the VSPs peer directly with each other or they peer through a third party VoIP Peering Provider (VPP). Essentially a VPP is like a VSP. It has the same logical functional components as a VSP and can also be deployed in composed or decomposed models. But a dedicated VPP is a better place to connect multiple VSPs and provide advanced peering services, including a privacy service. A simple illustration of VPP is shown in Figure 3. The use of VPP is particularly relevant in discussing service provider privacy. Although

VSP can deploy its own privacy services and achieve topology hiding, identity hiding of a VSP usually requires a third party privacy service, and VPP is the natural place to host this service. Therefore, in the following discussion we focus on a VPP-based VoIP peering architecture.
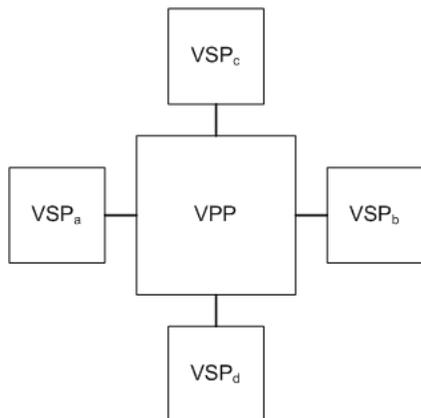
Figure 3: Peering through a VPP

A real VoIP peering network scenario is a result of the choices of the VoIP deployment models in Figure 1 and VSC-VSP connection methods in Figure 2, thus it could be in many different forms. To better understand that, we show an example in Figure 4. In this example scenario, VSCa connects to a cable service provider which provides both data and voice services (VSPa and IAPa). VSCb connects to a data service provider (IAPb) and a voice service provider (VSPb) separately. VSCb has a dedicated connection to VSPb. VSPa and VSPb both join the peering service provided by VPP. The signaling paths and media paths in this figure reveal several possible deployment models. At the VSPa side, all media traffic passes through an MF, resulting in the quasi-decomposed model of VSPa. At the VPP and VSPb, the MF element may or may not be used, resulting in either the quasi-decomposed model or the fully-decomposed model.

3.2 Applying Privacy Mechanisms to VoIP Peering

3.2.1 General Rules

Assuming we have a generic VPP-based VoIP peering architecture as in Figure 3 and the VPP implements the privacy mechanism described in Section 2, the VSP can either pre-arrange with the VPP for privacy services or dynamically request privacy services from the VPP as follows: If the VSP and IAP are collocated as in Figure 2(a), then regardless of the VSP deployment model, both signaling and media privacy need to be protected. The VSP should therefore request *header* privacy, *service provider* privacy, and *session* privacy from the VPP privacy service. If the VSP and IAP are separate as in Figure 2(b) and Figure 2(c), we should further examine the VSP deployment models. If composed or quasi-decomposed model as in Figure 1(a) and Figure 1(b) is used, both signaling and media pass through the VSP. Therefore, the VSP should request *header* privacy, *service provider* privacy, and *session* privacy from the VPP privacy service. If fully-decomposed model as in Figure 1(c) is used, the VSP only handles signaling, so it should request *header* privacy and *service provider* privacy from the VPP privacy service.

In short, if the VSP only handles signaling, then requesting *header* privacy and *service provider* privacy should be sufficient. If the VSP handles both signaling and media, then *header* privacy, *service provider* privacy and *session* privacy should all be requested. Note that the requirement to request
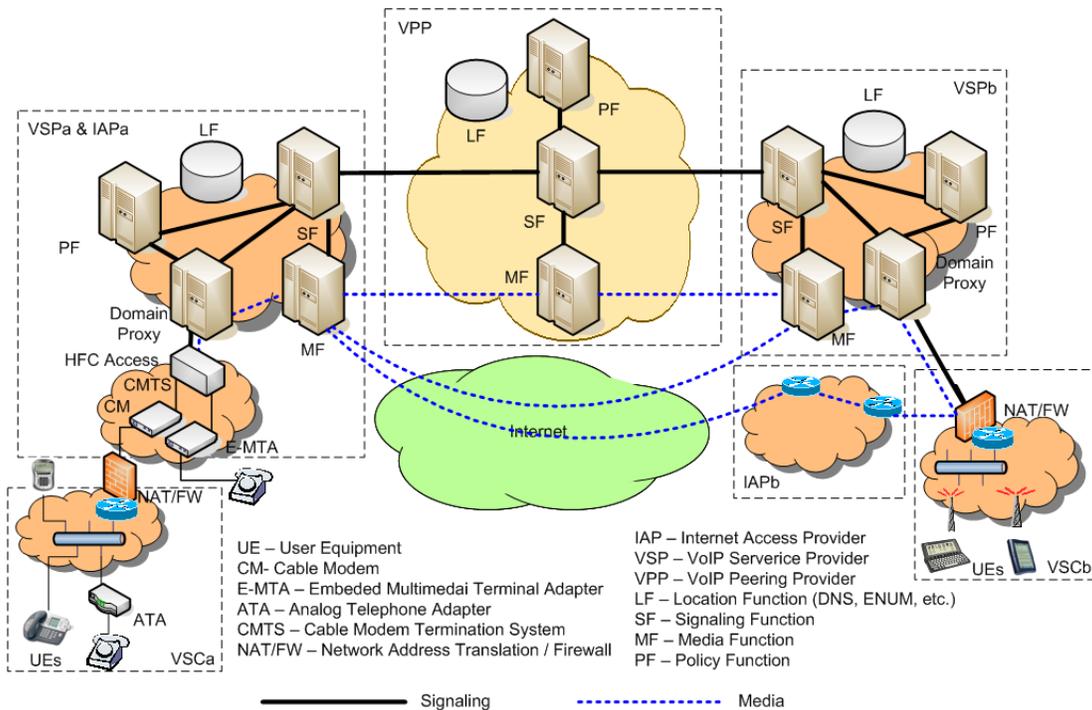
Figure 4: An example VoIP peering scenario

session privacy in the latter case could be relaxed if the service provider can afford some risk of its identity being guessed. Specifically, when *header* privacy with *service provider* privacy are available, an outsider knowing who the IAP is cannot conclude whether that provider is also the VSP for VoIP signaling, unless he already knows the fact that the customer's media and signaling go through the same provider.

Sometimes VSPs also implement their own network privacy functions, usually in their network border elements such as session border controllers. This case is an example of a service provider using a third party privacy service while keeping its own privacy service to perform part of the work, as we already discussed in Section 2.3.3.

In VoIP peering context, the VSP privacy requirement is usually bi-directional. This is relatively straightforward when the privacy service is provided by a VPP. As long as the VSPs for the caller and callee both follow the rules in Section 3.2.1 to pre-arrange privacy service or dynamically ask for appropriate privacy service levels in request and response messages going out of each of their domains, the VPP should be able to provide VSP privacy in both directions.

3.2.2 VoIP Peering Privacy Example Message Flow

Figure 5 through Figure 7 show the detailed message flow of a bi-directional privacy service example in VoIP peering; only the most important header fields are shown. In this example, caller Alice is served by VSPa (vspa.example.com) and callee Bob is served by VSPb (vspb.example.com). VSPa and VSPb peer with each other through VPP (vpp.example.com). VSPa and VSPb both want service provider privacy. Media paths are independent of the signaling paths. Therefore, both VSPa and VSPb request *service provider* and *header* privacy service levels from VPP. All privacy services are carried out at the VPP. The VPP privacy service implement the privacy mechanism presented in Section 2, including the "striping" method for Via, Route and Record-Route headers.

14

vspa.example.com   vpp.example.com   vspb.example.com
      proxy   proxy   proxy

Alice .... Bob

INVITE F1

INVITE: tel:+12029995678
v: ha.vspa.example.com
Route: vspa.example.com
f: +12125551234
  @vspa.example.com
t: tel:+12029995678
i: 1498765794
  @ha.vspa.example.com
m: +12125551234
  @ha.vspa.example.com

INVITE F2

INVITE: sip:+12029995678
  @vpp.example.com
v: vspa.example.com
v: ha.vspa.example.com
Record-Route:
  vspa.example.com
f: +12125551234
  @vspa.example.com
t: tel:+12029995678
i: 1498765794
  @ha.vspa.example.com
m: +12125551234
  @ha.vspa.example.com
Privacy: user, header

100 Trying F3

100 Trying
v: ha.vspa.example.com
f: +12125551234
  @vspa.example.com
t: tel:+12029995678
i: 1498765794
  @ha.vspa.example.com
m: +12125551234
  @ha.vspa.example.com

100 Trying F5

100 Trying
v: vspa.example.com
v: ha.vspa.example.com
f: +12125551234
  @vspa.example.com
i: 1498765794
  @ha.vspa.example.com
m: +12125551234
  @ha.vspa.example.com

INVITE F4

INVITE: sip:+12029995678
  @vspb.example.com
v: vpp.example.com
Record-Route:
  vpp.example.com
f: +12125551234
  @vpp.example.com
t: tel:+12029995678
i: 1498765794
  @9fxced86sl
m: +12125551234
  @vpp.example.com

100 Trying F7

100 Trying
v: vpp.example.com
f: +12125551234
  @vspa.example.com
t: tel:+12029995678
i: 1498765794
  @9fxced86sl
m: +12125551234
  @vpp.example.com

INVITE F6

INVITE: sip:+12029995678
  @hb.vspb.example.com
v: vspb.example.com
v: vpp.example.com
Record-Route:
  vspb.example.com
  vpp.example.com
f: +12125551234
  @vpp.example.com
t: tel:+12029995678
i: 1498765794
  @9fxced86sl
m: +12125551234
  @vpp.example.com

Figure 5: Example VoIP peering privacy service message flow - part I

vspa.example.com proxy ⋯⋯ vpp.example.com proxy ⋯⋯ vspb.example.com proxy

Alice ⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯⋯ Bob

180 Ringing F8

180 Ringing
v: vspb.example.com
v: vpp.example.com
Record-Route:
  vspb.example.com
  vpp.example.com
f: +12125551234
  @vpp.example.com
t: tel:+12029995678
i: 1498765794
  @9fxced86sl
m: +12029995678
  @vspb.example.com

180 Ringing F9

180 Ringing
v: vpp.example.com
Record-Route:
  vspb.example.com
  vpp.example.com
f: +12125551234
  @vpp.example.com
t: tel:+12029995678
i: 1498765794
  @9fxced86sl
m: +12029995678
  @vspb.example.com
Privacy: user, header

180 Ringing F10

180 Ringing
v: vspa.example.com
v: ha.vspa.example.com
Record-Route:
  vpp.example.com
  vpsa.example.com
f: +12125551234
  @vspa.example.com
t: tel:+12029995678
i: 1498765794
  @ha.vspa.example.com
m: +12029995678
  @vpp.example.com

180 Ringing F11

180 Ringing
v: ha.vspa.example.com
Record-Route:
  vpp.example.com
  vpsa.example.com
f: +12125551234
  @vspa.example.com
t: tel:+12029995678
i: 1498765794
  @ha.vspa.example.com
m: +12029995678
  @vpp.example.com

200 OK F12

200 OK
v: vspb.example.com
v: vpp.example.com
Record-Route:
  vspb.example.com
  vpp.example.com
f: +12125551234
  @vpp.example.com
t: tel:+12029995678
i: 1498765794
  @9fxced86sl
m: +12029995678
  @vspb.example.com

200 OK F13

200 OK
v: vpp.example.com
Record-Route:
  vspb.example.com
  vpp.example.com
f: +12125551234
  @vpp.example.com
t: tel:+12029995678
i: 1498765794
  @9fxced86sl
m: +12029995678
  @vspb.example.com
Privacy: user, header

200 OK F14

200 OK
v: vspa.example.com
v: ha.vspa.example.com
Record-Route:
  vpp.example.com
  vpsa.example.com
f: +12125551234
  @vspa.example.com
t: tel:+12029995678
i: 1498765794
  @ha.vspa.example.com
m: +12029995678
  @vpp.example.com

200 OK F15

200 OK
v: ha.vspa.example.com
Record-Route:
  vpp.example.com
  vpsa.example.com
f: +12125551234
  @vspa.example.com
t: tel:+12029995678
i: 1498765794
  @ha.vspa.example.com
m: +12029995678
  @vpp.example.com

ACK F16

ACK: +12029995678
  @vpp.example.com
v: ha.vspa.example.com
Route:
  vspa.example.com
  vpp.example.com
f: +12125551234
  @vspa.example.com
t: tel:+12029995678
i: 1498765794
  @ha.vspa.example.com
m: +12125551234
  @ha.vspa.example.com

ACK F17

ACK: +12029995678
  @vpp.example.com
v: vspa.example.com
v: ha.vspa.example.com
Route:
  vpp.example.com
f: +12125551234
  @vspa.example.com
t: tel:+12029995678
i: 1498765794
  @ha.vspa.example.com
m: +12125551234
  @ha.vspa.example.com
Privacy: user, header

ACK F18

ACK: +12029995678
  @vspb.example.com
v: vpp.example.com
Route:
  vspb.example.com
f: +12125551234
  @vpp.example.com
t: tel:+12029995678
i: 1498765794
  @9fxced86sl
m: +12125551234
  @vpp.example.com

ACK F19

ACK: +12029995678
  @hb.vpp.example.com
v: vspb.example.com
v: vpp.example.com
f: +12125551234
  @vpp.example.com
t: tel:+12029995678
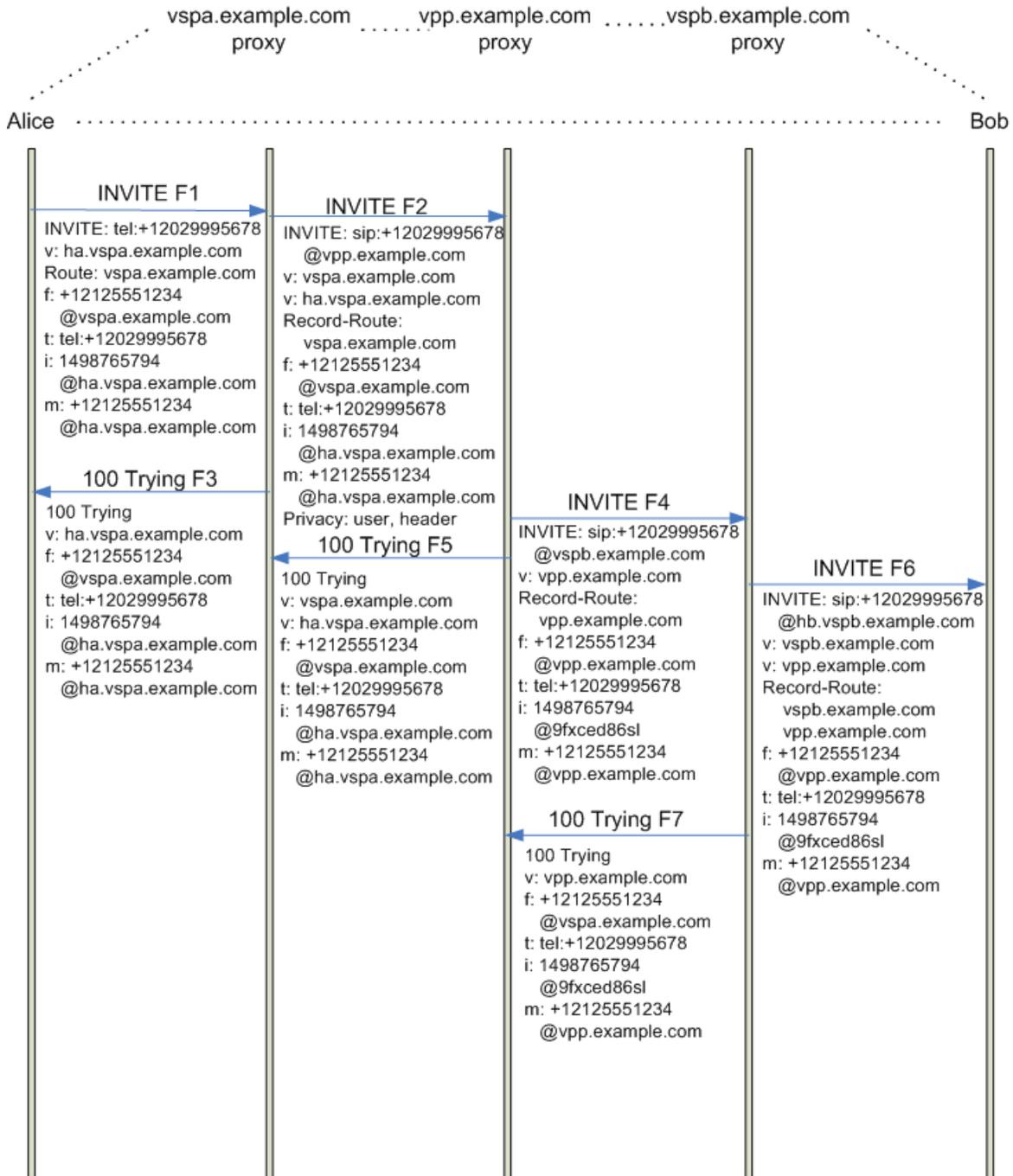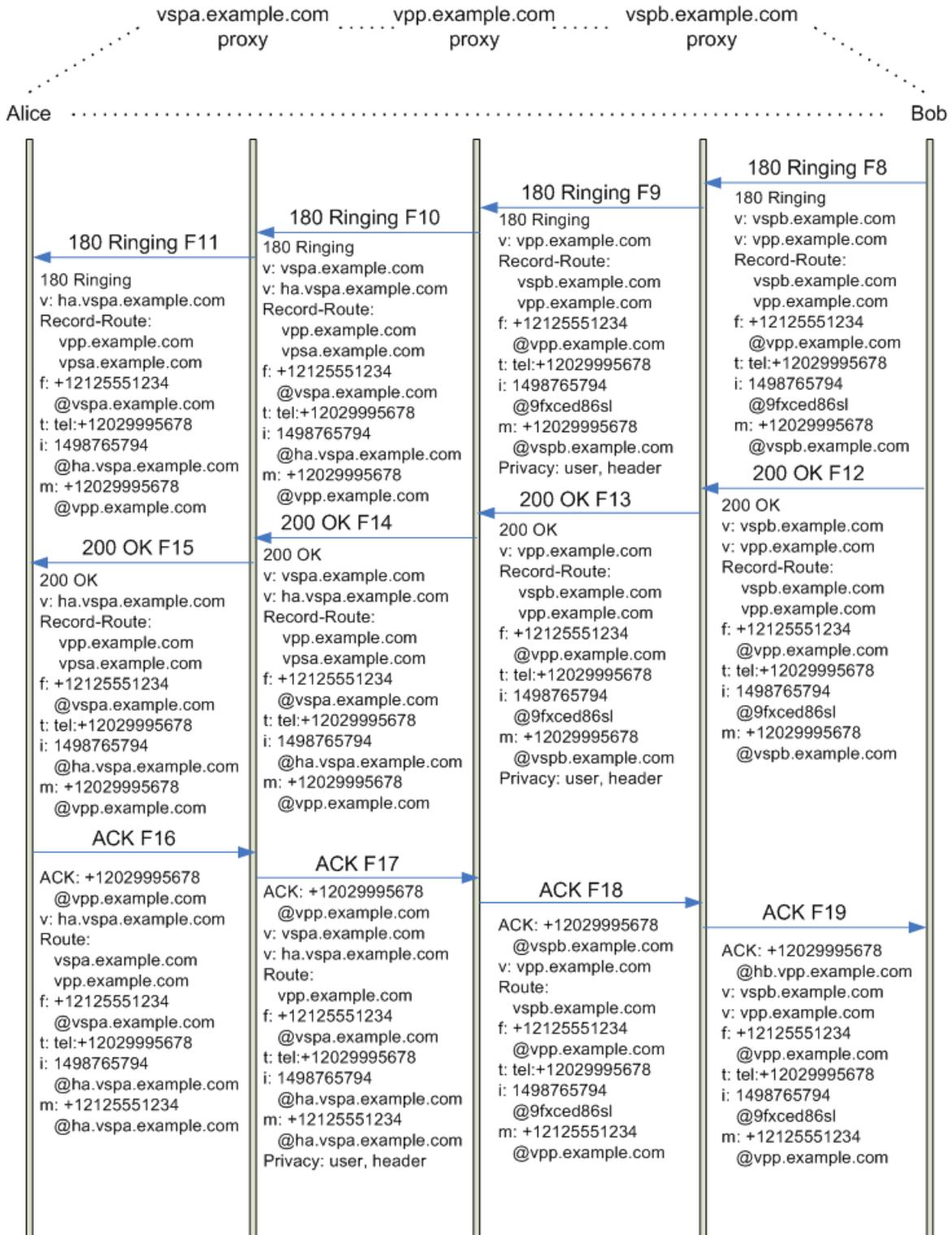i: 1498765794
  @9fxced86sl
m: +12125551234
  @vpp.example.com

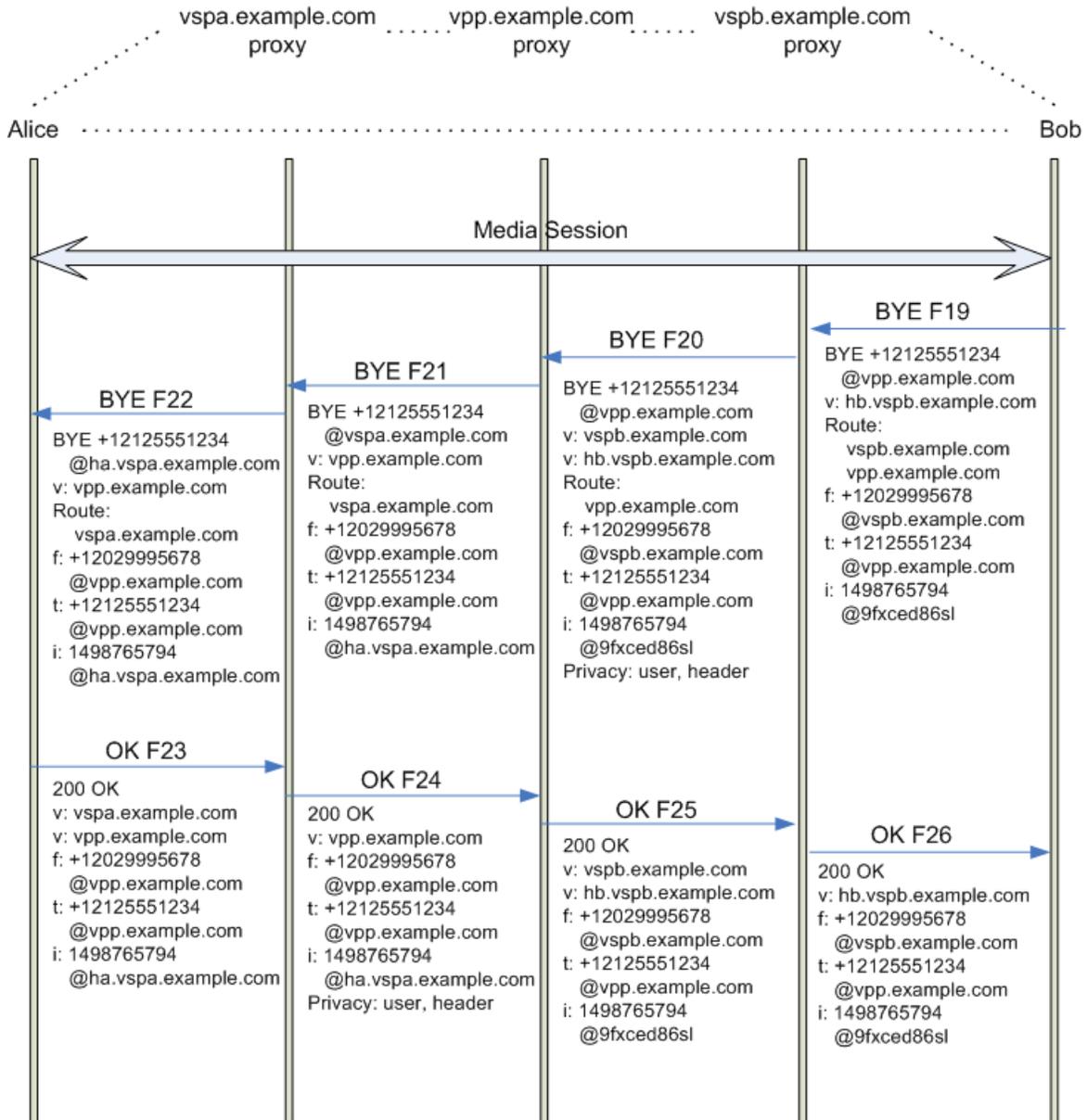Figure 6: Example VoIP peering privacy service message flow - part II

Figure 7: Example VoIP peering privacy service message flow - part III

4 Conclusions

Existing work on SIP privacy focuses on privacy-sensitive information for end users. The VoIP peering community needs a privacy mechanism that also cover privacy-sensitive information for VSPs. Specifically, VSPs want to prevent their identity and topology information from being exposed to unintended parties during the VoIP communication. In this report, we summarize and extend related work, and present a unified VoIP privacy mechanism for both user and service providers. We first examined the SIP, SDP, RTP and RTCP message fields that potentially reveal user or service provider information. Then we discussed privacy service functions that can be performed by the user side and the network intermediaries. The user side can apply removal, anonymization or encryption to those privacy-sensitive message fields. The network intermediaries can perform similar functions on behalf of the user side if necessary. In addition, the network intermediaries also need to handle privacy for several routing specific fields, which cannot be done at the user side. The different network privacy functions can be indicated by the SIP Privacy header values. We illustrated the use of appropriate Privacy header values to achieve user level and service provider level privacy requirements. As an example, we discussed a generic VoIP peering architecture characterized by different options of VoIP deployment and connection models, and show how the VSPs can achieve identity and topology hiding in VoIP peering context using the proposed privacy mechanism.

5 Acknowledgements

References

[1] J. Peterson, "A Privacy Mechanism for the Session Initiation Protocol (SIP)," RFC 3323 (Proposed Standard), Nov. 2002.

[2] C. Jennings, J. Peterson, and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks," RFC 3325 (Informational), Nov. 2002.

[3] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," RFC 3261 (Proposed Standard), June 2002, Updated by RFCs 3265, 3853, 4320.

[4] M. Baugher, D. McGrew, M. Naslund, E. Carrara, and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)," RFC 3711 (Proposed Standard), Mar. 2004.

[5] M. Handley, H. Schulzrinne, E. Schooler, and J. Rosenberg, "SIP: Session Initiation Protocol," RFC 2543 (Proposed Standard), Mar. 1999, Obsoleted by RFCs 3261, 3262, 3263, 3264, 3265.

[6] B. Byerly, D. Daiker, and S. Bhatnagar, "SIP Record-Route/Route Hiding," Internet draft, Oct. 2000, work in progress.

[7] H. Schulzrinne, "The tel URI for Telephone Numbers," RFC 3966 (Proposed Standard), Dec. 2004.

[8] J. Rosenberg, "Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP)," Internet draft, July 2006, work in progress.

[9] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 3550 (Standard), July 2003.

[10] B. Ramsdell, "Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification," RFC 3851 (Proposed Standard), July 2004.

[11] R. Sparks, "Internet Media Type message/sipfrag," RFC 3420 (Proposed Standard), Nov. 2002.

[12] R. Penno (Editor), "SPEERMINT Peering Architecture," Internet draft, Aug. 2006, work in progress.