A Survey of SIP Peering

Lars Strand
lars.strand@nr.no

Norwegian Computing Center, P.O. Box 114 Blindern, NO-0314 Oslo, Norway

Wolfgang Leister
wolfgang.leister@nr.no

Norwegian Computing Center, P.O. Box 114 Blindern, NO-0314 Oslo, Norway

Abstract

When placing a call from one SIP Service Provider to another, the call is traditionally routed over PSTN, instead of IP. This can lead to higher costs, reduced quality, and lack of functionality. These issues can be addressed by setting up a SIP peer between providers. A SIP peer is a layer 5 interconnection between two SIP Service Providers for the purpose of routing real-time and quasi-real-time call signalling between their customers. We survey the SIP peering architecture and show security implications.

Keywords: VoIP, SIP, security, peering, PSTN

1. Introduction

Conventional telephony, also called Plain Old Telephone Service (POTS), still accounts for the majority of telephony calls. POTS uses a circuit switched network, where each call establishes a dedicated circuit between two nodes with fixed bandwidth before communicating. These circuit switching networks form what is called the Public Switched Telephone Network (PSTN).

In contrast to PSTN, Voice over IP (VoIP) uses packet switching for sending data. A packet switching network divides the traffic into a sequence of packets that are sent over a shared network. VoIP is considered the emerging technology that will eventually take over from PSTN. In the 1970s, experiments with transmitting voice over IP networks was conducted [1]. However, it was not until the mid 1990s that the H.323 protocol [2] and the Session Initiation Protocol (SIP) [3] were standardised and widely deployed. Today, SIP is the preferred signalling protocol in the VoIP industry.

SIP has evolved into a mature and stable standard, and is rapidly being adopted for VoIP applications by the industry. However, in contrast to the original plans of the designers of SIP, the VoIP providers do not use the so-called open *email model* for communication in-between them due to security concerns. As a result, providers have started to set up ad hoc SIP peering. However, such individual solutions are not scalable. Therefore, we review initiatives to organise SIP peering.

The rest of the paper is organised as follows: We give an introduction to SIP and the *email model* in Section 2. We explain SIP peering, its architecture and logical functions in Section 3. In Section 4, we discuss security concerns and investigate whether SIP peering solves any of these. We then summarise and try to identify which areas require more work in Section 5 and give an outlook in Section 6.

2. SIP and the email model

SIP is an application layer protocol that handles multimedia sessions. It is used to negotiate, establish, change and tear-down the *context* of a multimedia flow; other protocols, such as the Real-time Transport Protocol (RTP), are used for the media (voice) transport [4].

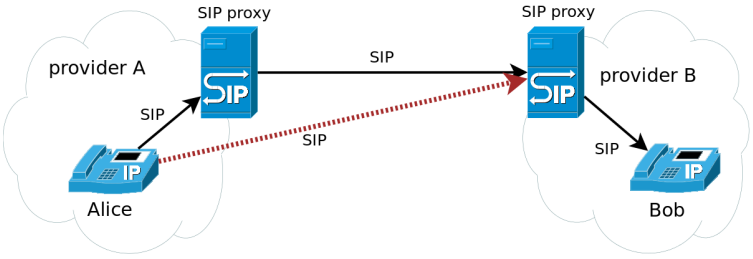

Figure 1: When Alice calls Bob, the message path form a trapezoid going from Alice's UA through provider A's SIP proxy which forwards the call to provider B's proxy and then to Bob.

Illustrated in a simple example, SIP works as follows: When Alice calls Bob, as depicted in Figure 1, a SIP `INVITE` message is sent from Alice's User Agent (UA), via one or more SIP proxies, to Bob's UA. Before allowing Alice to send an `INVITE` request to Bob, Alice's SIP proxy may request Alice to authenticate and do rudimentary SIP header checks before forwarding to Bob's SIP proxy. The media (RTP) might take a direct path between Alice and Bob, thus forming a "SIP trapezoid" for the message path. However, Alice may send an SIP `INVITE` directly to Bob's SIP proxy. Provider B's SIP proxy cannot distinguish whether a request arrived from a SIP proxy or a UA directly. This is depicted in red in Figure 1.

To address Bob, Alice uses Bob's SIP address (URI). A SIP URI is structured in the same way as email addresses with *username@domain*. While a call is routed to the destination, only the domain part is relevant. The username is only interpreted by the SIP server in the receiver's domain [5]. The global public DNS is used to map the domain part to one or more SIP ingress servers that handle incoming SIP requests. We denote this mode of routing and addressing SIP as the *email model*.

SIP has not seen global reachability as outlined in the SIP standard [6]. There are three main reasons why the *email model* for SIP has failed. *1*) The telephony providers have traditionally collected termination fees between communication partners (other providers). If everyone directly is able to connect to everyone, no business relationships between providers are necessary. Therefore, the carriers have no economic incentive to switch to a global reachable SIP addressing scheme. *2*) Operators of public telephony services need to comply to a range of legal regulatory requirements. These requirements are applicable for the PSTN with clear boundaries between telephony operators and telephony users. *3*) There are a range of security concerns to which no simple solution exists:

(a) Unwanted calls, also known as "Spam over Internet Telephony" (SPIT), are a threat to the VoIP infrastructure. Since there are currently few open SIP servers, SPIT has not yet grown to be a widespread problem compared to *email spam*. Note that problems related to spam emerged after the number of open SMTP[1] servers increased. SPIT is harder to prevent than email spam, since VoIP calls are interactive — the content or the intentions of a call are not identifiable in advance, before the receiver picks up the phone. Therefore, filters for SPIT cannot be applied as easily as spam filters for email. Also social factors are important, since one usually picks up the phone when it rings, instead of being able to choose when and how often to check email. Providers fear that SPIT could become common if they open up their SIP services to the Internet [7].

(b) Assuring the *identity* of the caller. Signalling in PSTN has traditionally been trusted between carriers, and by end users (caller-id). This trust is not applicable to open SIP servers, since the INVITE message can come from any user on the Internet and the caller-id can easily be spoofed. Different authentication mechanisms for SIP exist like S/MIME [8], the "Asserted Identity" extension [9] and the "Identity" header extension [10]. These will be discussed in Section 4.2. Unfortunately, these identity mechanisms have not been deployed nor supported to a great extent worldwide [11].

[1]Simple Mail Transfer Protocol (SMTP) is the Internet standard for email.

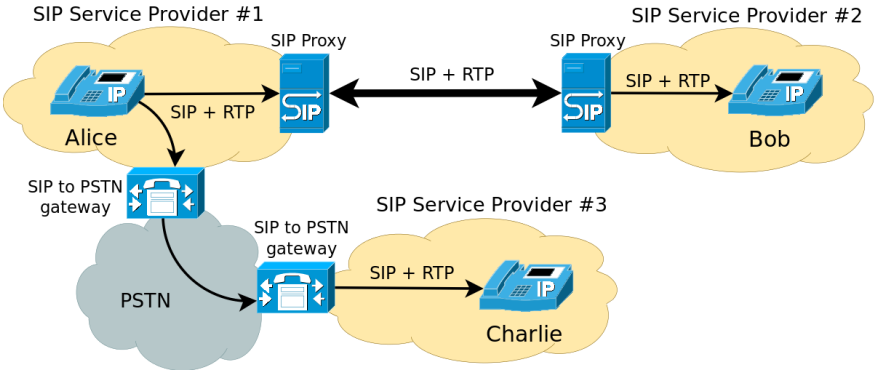

Figure 2: The two scenarios: When Alice calls Bob, SIP peering is used. Without SIP peering between two SSPs, calls are routed through the PSTN, as illustrated where Alice calls Charlie.

(c) *Denial of Service* attacks are threats to availability. The *email model* is particularly vulnerable to DoS attacks, since SIP servers need to accept request from anyone on the Internet. This makes it hard to guarantee a stringent Quality of Service (QoS) agreement to customers and other SIP providers.

As a result of these concerns, SIP Service Providers (SSP) have not deployed open SIP servers (the *email model*). Therefore, when a call is destined for outside the SSPs domain, it is routed over PSTN to the provider network of the receiver, as outlined in Figure 2. This makes it necessary to transcode both content and signalling of the call. Routing calls over PSTN has disadvantages:

1. Managing VoIP-to-PSTN gateways adds administrative overhead, extra hardware costs, and extra resources to configure, deploy and manage the gateways on a daily basis.

2. Sending calls over PSTN is more expensive for the VoIP provider, since the provider must pay termination fees to the PSTN provider.

3. A call traversing the VoIP-to-PSTN gateway needs to be transcoded from VoIP before going into the PSTN. If the receiving end, the callee, also uses VoIP, the call must be transcoded from PSTN to VoIP. This is illustrated in Figure 2 in the scenario where Alice calls Charlie. Even if both VoIP and PSTN use the G.711 voice codec [12], delays are possible, and information can get lost during transcoding. The use of "wideband" speech codecs like G.722 [13] does not give any advantages. These wideband codecs provide superior voice quality and have the potential to become a differentiator for VoIP [14].

4. PSTN does not carry services that are offered by SIP, such as video, IM and presence. Therefore, these services cannot be offered to the customers as a service between providers.

To overcome these disadvantages without deploying open SIP servers, the SSPs intend to set up SIP peering between each other. This eliminates the need for transcoding to/from PSTN. Unfortunately, there is no standard nor suitable best practices for how to set up these peers between the SSPs.

There have been some industry attempts to create SIP peering recommendations [15], but no defining standards. The Internet Engineering Task Force (IETF) has acknowledged this, and created the SPEERMINT[2] Working Group (WG) with the goal to identify architecture requirements, discuss security considerations, and define best practises for SIP peering.

[2]The charter of the Session PEERing for Multimedia INTerconnect (SPEERMINT) WG is available at http://www.ietf.org/dyn/wg/charter/speermint-charter.html.

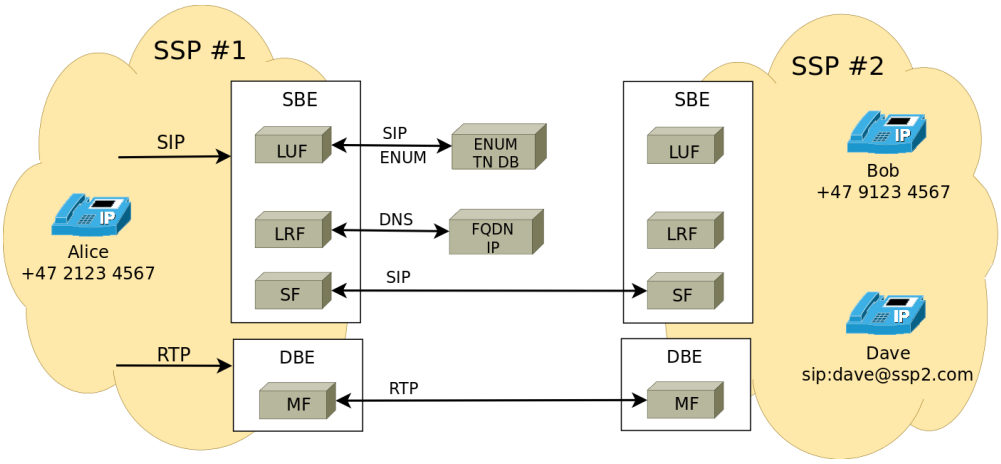

Figure 3: Logical functions defined for the SIP peering architecture.

3. SIP peering

The term "Voice over IP peering" or just "Voice peering" refers to a wide range of practices pertaining to the interconnections of VoIP service providers. "SIP peering" refers to Voice peering, taking into account how SIP can be used in a secure and standardised manner for interconnections between VoIP service providers.

An important distinction must be made between the traditional peering on the IP layer (Layer 3 in the OSI protocol stack [16]), and peering on the application layer (OSI layers 5–7). SIP is an application layer protocol designed to run independently of the transport layer (TCP/UDP/SCTP). Therefore, SIP peering interconnection operates on the application layer, and assumes that lower layer functionality, like IP routing, are handled by other network processes.

Rather than introducing new SIP extensions, SIP peering uses existing protocol standards (SIP, RTP, ENUM, DNS) as building blocks, to create a set of best practices and operational procedures. A number of logical functions have been defined as part of the SIP peering architecture, which we briefly describe.

3.1. The SPEERMINT Architecture

The SPEERMINT architecture, shown in Figure 3, consists of two logical functions, the Signalling path Border Element (SBE) and the Data path Border Element (DBE). The SBE provides the signalling functionality to and from peering partners, i.e., the SSPs. The DBE provides media-related functions, such as firewall-traversal support and transcoding.

The *SBE* consists of a number of logical functions, but does not redefine how SIP uses input and output variables to create Session Establishment Data (SED). *SED* denotes a set of parameters that the outgoing SBE need to complete a call. These parameters may include SIP addresses (SIP URIs), the address of the ingress SIP proxy including the fully qualified domain name (FQDN), port and transportation method (TCP/UDP/TLS), and security parameters (TLS certificate data).

For the SBE to acquire the parameters needed to complete a call, it relies on two functions: the *Look-Up Function* (LUF) and the *Location Routing Function* (LRF). The LUF is used if the destination is a telephone number, and we need to translate the number to a routable Internet SIP address. This is done by using *ENUM* [17, 18] which translates public telephone numbers [19] to SIP addresses (SIP AOR) by DNS.

After the LUF has provided a SIP address, the SBE needs to determine the target domain to which the request should be routed. This lookup function is performed by the LRF, and consists of an ordinary DNS lookup of the target domain. The result of the DNS lookup will provide to the *Signalling Function* (SF) the necessary SED parameters needed to address and find the target domain (ingress point). The SF, usually a SIP proxy, can then perform routing of the request to the correct destination.

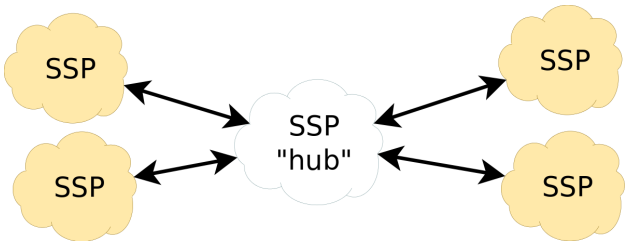

Figure 4: A "hub-based federation" where SSPs form a star peering topology.

After the signalling phase (SIP handshake) has been completed, the actual content (voice/video) session is established using the DBE. The DBE consists of one logical function, the *Media Function* (MF), which is responsible for the actual delivery of multimedia communication between users and providers.

3.2. Peering arrangements

SIP peering is classified into *static* and *on-demand*, and each of these into *direct* and *indirect* peering. The adoption of on-demand peering has seen low industrial penetration [20].

For *static direct peering* two SSPs have agreed on a peering relationship before the exchange of phone calls, and the sending SSP reaches the receiving SSP in one hop. For *static indirect peering* signalling and media path must be established via one or more SSPs before reaching the destination SSP. A group of SSPs that peer with each other is called a *federation*.

There is an administrative overhead to exchange and keep the various lookup-function data up to date between the SSPs. Therefore, if each SSP would peer with every other SSP, the number of peering agreements would be infeasibly high. An emerging solution consists of using one single SSP as a *hub* for the other SSPs. This hub SSP provides an assisted LUF/LRF for the other SSPs as well as a central peering point, thus forming a star topology as shown in Figure 4. This hub-based federation model is currently being adopted by the industry, delegating LUF/LRF and administrative tasks, like abuse handling or technical requirements for the peers, to the central hub.

4. Security considerations

The SIP core specification states that *"SIP is not an easy protocol to secure"* [3, page 232]. This is because SIP has been extended extensively with additional functionality for different targets [4], and the number of components involved in a typical SIP based VoIP setup ranges from user devices (UAs) to SIP servers (proxies and registration), firewalls, gateways, and supporting servers like LUF/LRF lookup. With this diversification, different security requirements and mechanisms apply.

A clear and concise VoIP threat taxonomy is given by VOIPSA [21]. VoIP threats are discussed by Keromytis [22] and the wide range of VoIP security threats have stimulated research in this area [23]. Several security mechanisms for countermeasures have been proposed, but no single security mechanism is suited to address all these security threats concerning SIP [24, 25, 26]. While there are many threats for VoIP, we consider *Spam over Internet Telephony* (SPIT), weaknesses of *authentication*, and *Denial of Service* (DoS) attacks as the most relevant from the perspective of SIP peering.

4.1. SPIT, SPIM and SPPP

As spam is a problem for the email infrastructure, it is expected that the VoIP counterparts of spam will emerge for the VoIP infrastructure. The amount of email traffic that is classified as spam compared to ordinary email is estimated to be around 90% according to the Spamhaus project[3]. For an end-user, it

[3]The Spamhaus Project: http://www.spamhaus.org

usually takes a couple of seconds to classify an email as spam, and delete it. However, the costs on a global scale can be significant, when adding up the amount of time each user uses on handling spam. Additionally, email providers must invest in anti-spam measures and develop routines to handle spam.

For VoIP, the spam problem is related to vulnerabilities of SIP, and to the fact that VoIP, in contrast to email, is used interactively. Spam over SIP can be classified in three groups [27]: *1*) Spam over Internet telephony (SPIT) is defined as unwanted calls. Here, the *spitter* either plays back a pre-recorded message, or does telemarketing. *2*) Spam over instant message (SPIM) is defined as sending unsolicited instant messages. SPIM has similar properties as email spam. *3*) Spam over presence protocol (SPPP) is defined as sending bulks of presence requests (SIP `SUBSCRIBE`). If a user accepts such a request, the spammer is usually is put on the user's "buddy list", and is consequently allowed to send SPIM or SPIT.

Of these, SPIT is considered the most attractive and effective for spammers [24]. SPIT differs from email spam in two distinct ways: First, there are social norms and behaviour. A user can choose when to check email, but when the phone rings, the call usually is answered. It will take us a couple of seconds to classify the call as SPIT, but then we have already been disrupted. If this happens often it is considered very disruptive. Second, preventing SPIT is harder than email spam, since the content (media) is not available until after the user picks up the phone. This differs from email spam, where the content can be filtered and classified in advance.

There is a low amount of SPIT today, since the number of open SIP servers is limited. However, once the number of VoIP users increases above a certain limit, this could change [6]. Several anti-SPIT solutions have been proposed. The SPIDER research project provided an extensive analysis of SPIT, and presents different anti-SPIT measures[4]. The RFC 5039 [27] analyses spam over SIP, and discusses whether anti-spam solutions for email are applicable for SPIT.

Both the SPIDER project and RFC 5039 conclude that there are no simple and clear solutions to avoid SPIT completely. The SPIT problem would have been less significant if these threats had been taken into account already during the specification phase of SIP. Improved SIP authentication methods, and assuring a caller's identity across SSPs, are necessary to prevent SPIT. However, it might be problematic to improve authentication methods, after the industry adoption of SIP.

4.2. Authentication

A definition of authentication is *"the binding of one identity to a subject"* [28]. The user has several expectations that are related to authentication. For example, the caller will expect a call to be established with the intended callee. The callee will expect to talk to the person that the caller claims to be. There are several authentication mechanisms in SIP, some are mandatory and others are SIP extensions that have not seen widespread adaption.

Within one administrative domain (usually within the domain of an SSP), the SIP proxies can demand the clients (UAs) to authenticate themselves before use. But there are no obligations do to so. Some providers only authenticate calls that are destined to PSTN, since the providers pay interconnection fees when routing through PSTN [24]. A UA can also avoid authentication by contacting the callee's SIP proxy directly, as shown in Figure 1, where Alice contacts provider B's SIP proxy without going through provider A's SIP proxy. Both the caller and the callee should be mutually authenticated.

SIP provides several authentication mechanisms, but only the Digest Access Authentication method is mandatory:

- The Digest Access Authentication (DAA) [3, section 22] is the most common authentication method, since its support is mandatory. Unfortunately, DAA provides only weak authentication, and is vulnerable to a series of attacks, including off-line dictionary attacks and registration attacks [11].

- A more secure authentication method is achieved by encapsulating the SIP message into a Secure MIME (S/MIME) format [8]. The encapsulated SIP message can be signed or encrypted, or both. The S/MIME content is carried in the payload of a new outer SIP message. Since S/MIME relies on

[4]Spam over Internet telephony detection service (SPIDER): http://www.projectspider.org/

certificates, each UA must obtain and install an individual certificate from a Certification Authority (CA) before use. Since no single CA is trusted by all UAs across the SSPs, a UA must have support for multiple root certificates. This, and other certificate handling issues like revoking and renewing, complicates the usage of certificates. The industry support for S/MIME has been limited so far.

- SSPs require that the caller's identity can be assured, to comply to regulatory requirements such as the ability to trace back a call. Since the "`From`" SIP header field easily can be manipulated, there is a need to assert the caller's identity between SSPs. This can be achieved by including an asserted-identity SIP header [9]. After a UA has been authenticated against the local proxy using DAA, the proxy adds a "`P-Asserted-Identity`" SIP header. This identity header is sent in clear, and is not protected by cryptography in any way. Since an attacker could use reply- and modification attacks, or remove the header altogether, this method is limited to trusted domains where SIP proxies communicate over trusted links.

- Another approach introduces two new SIP headers and a SIP "authentication service" [10]. After the UA has been authenticated to its SIP proxy, the originating SIP proxy signs a hash over some particular SIP headers, and includes the signature as an "`Identity`" header. Also included is an "`Identity-Info`" header, which contain an URI for the caller's certificate. The SIP server in the callee's domain computes the same hash, and compares the result from the originating proxy. This authenticates the caller (or more precisely the caller's proxy) but not the callee. An attacker might remove these headers in transit without an implication for the callee.

Within one SSP, the use of DAA in combination with TLS can authenticate, with confidence, the identity of the UA. But within a federation of SIP peers, no direct mutual authentication exists between UAs of different SSPs. We can only achieve transitive trust between SSPs, and must hope that there are no weak links in this chain.

4.3. Denial of Service

A Denial of Service (DoS) attack affects the availability of a service. For VoIP, this means that legitimate voice communication could be prevented from an attacked service. A *Distributed DoS* (DDoS) attack is when several nodes are involved in the attack, and are one of the most common network attacks on the Internet. A DDoS attack is rather simple to achieve, and effective, while it is hard to protect a service against it. There are three categories of DoS attacks: *1*) The most common DoS attack involves flooding the victims node/service with network traffic, thereby exhausting its resources such as memory, CPU, bandwidth, or otherwise. *2*) A misuse attack intentionally misuses or malforms SIP messages to interrupt or terminate a VoIP call or service resulting in a denied availability for legitimate users. An example is SIP call hijacking [29]. *3*) An indirect attack, is an attack on supporting infrastructure services that VoIP rely on. For example, an attack on the LUF/LRF services would reduce the availability of the VoIP service, since SIP address resolution would not be available.

All components in a VoIP installation are vulnerable to DoS attacks. Attacking an SBE has thus more far-reaching consequences than an attack on a single UA. With the adoption of "federated hub peering", a successful DoS attack against the hub SSP will seriously degrade the availability of inter-SSP VoIP calls, since it can be considered as a single point of failure.

5. Challenges for SIP Peering

The rationale behind SIP peering is to achieve global and universal VoIP connectivity, without using the *email model* in inter-SSP traffic. The peering architecture needs to build a global federation that handles trust. In addition to trust between the SSPs, the trust relationship between the SSPs and their customers is important.

While trust and security concerns can be solved on a small scale, the global VoIP infrastructure needs to be scalable. It is a challenge to scale SIP peering to a global level without exposing LUF and LRF data to

the world. It is also important to be able to route VoIP calls effectively through federations of federations. In the case of technical problems with a VoIP service, it is necessary to access diagnostics, and be able to trace calls, in order to figure out which network is causing problems. The challenge of receiving suitable diagnostics is not specific to SIP peering, but SIP peering amplifies this need.

Security considerations related to SIP are also applicable in a peering relationship. But to demand that the SSPs use and enforce network security mechanisms like TLS/IPSec and deploy secure and scalable network design everywhere is unrealistic. Focus should be on enhancing the existing security mechanisms already defined in SIP.

Also, more detailed work is needed on authentication. The only mandatory authentication method is DAA, which is vulnerable to several attacks. Improving this authentication method would counter the most rudimentary attacks. When using DAA, a UA is authenticated locally within a SSP, but this authentication is not assured across multiple SSPs. SPEERMINT mandates a trust relationship between peering SIP proxies, but the available authentication methods do no provide mutual authentication between UAs. Other approaches should be investigated, like reputation based systems or multi-factor authentication systems.

6. Outlook

While SIP is an established standard within VoIP, we have seen that the *email model* is not a good solution for signalling between SSPs due to security concerns. SIP peering has emerged as one solution. However, SIP peering will only provide a solution if the SSPs adhere to the recommendations of the SPEERMINT WG, instead of developing their own non-standard or ad-hoc peering solutions. Besides security, issues of scalability and service quality need to be addressed.

Interconnecting all SSPs in one global SIP peering mesh (federation) is unfeasible. Even one global federation of federation is impractical and exposed to huge challenges, such as SIP routing. Also, demanding support for roaming between SSPs, adds to the complexity of any solution.

Acknowledgement

This research is funded by the EUX2010SEC project in the VERDIKT framework of the Norwegian Research Council (Norges Forskningsråd, project 180054). The authors want to thank Josef Noll who inspired us for this survey. We also thank Trenton Schulz for discussions while preparing this paper and Arne-Kristian Groven for comments on the earlier drafts of this paper.

References

[1] D. Cohen, Specifications for the Network Voice Protocol (NVP), RFC 741, 1977.
[2] International Telecommunication Union, H.323: Packet based multimedia systems, ITU-T Recommendation H.323, 2006.
[3] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, SIP: Session Initiation Protocol, RFC 3261 (Proposed Standard), 2002. Updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621, 5626, 5630, 5922, 5954, 6026, 6141.
[4] H. Sinnreich, A. B. Johnston, Internet communications using SIP: Delivering VoIP and multimedia services with Session Initiation Protocol, John Wiley & Sons, Inc., New York, NY, USA, second edition, 2006.
[5] J. Rosenberg, H. Schulzrinne, Session Initiation Protocol (SIP): Locating SIP Servers, RFC 3263 (Proposed Standard), 2002.
[6] O. Lendl, VoIP Peering: Background and Assumptions, Technical Report, IETF, 2008.
[7] J. Rosenberg, SIP Peering: What It Does and Doesn't Mean, SIP Magazine Speaking SIP (2006).
[8] J. Peterson, S/MIME Advanced Encryption Standard (AES) Requirement for the Session Initiation Protocol (SIP), RFC 3853 (Proposed Standard), 2004.
[9] C. Jennings, J. Peterson, M. Watson, Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks, RFC 3325 (Informational), 2002. Updated by RFC 5876.
[10] J. Peterson, C. Jennings, Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP), RFC 4474 (Proposed Standard), 2006.
[11] A. M. Hagalisletto, L. Strand, Formal modeling of authentication in SIP registration, in: Second International Conference on Emerging Security Information, Systems and Technologies SECURWARE '08, IEEE Computer Society, 2008, pp. 16–21.
[12] International Telecommunication Union, Pulse Code Modulation PCM of Voice Frequencies, ITU-T Recommendation G.711, 1993.

[13] International Telecommunication Union, 7 kHz Audio-Coding within 64 kbits/s, ITU-T Recommendation G.722, 1993.
[14] M. Miller, The VoIP Peering Puzzle, Enterprise VoIP Planet (2006).
[15] C. Sibley, C. Gatch, (eds), SIPconnect 1.0 Technical Recommendation, Technical Report, SIP Forum, 2008.
[16] H. Zimmermann, OSI Reference Model–The ISO Model of Architecture for Open Systems Interconnection, IEEE Transactions on Communications 28 (1980) 425–432.
[17] P. Faltstrom, M. Mealling, The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM), RFC 3761 (Proposed Standard), 2004. Obsoleted by RFCs 6116, 6117.
[18] J. Peterson, enumservice registration for Session Initiation Protocol (SIP) Addresses-of-Record, RFC 3764 (Proposed Standard), 2004. Updated by RFC 6118.
[19] International Telecommunication Union, The International Public Telecommunication Numbering Plan, ITU-T Recommendation E.164, 2005.
[20] A. Uzelac, Y. L. Lee, IETF DRAFT: VoIP SIP Peering Use Cases, Technical Report, IETF, 2010.
[21] VoIPSA, VoIP security and privacy threat taxonomy, Public Realease 1.0, 2005.
[22] A. D. Keromytis, Voice over IP: Risks, Threats and Vulnerabilities, in: Proceedings of the Cyber Infrastructure Protection (CIP) Conference, New York.
[23] A. D. Keromytis, A Survey of Voice Over IP Security Research, in: Proceeding of the 5th International Conference on Information Systems Security (ICISS), pp. 1 – 17.
[24] D. Sisalem, J. Floroiu, J. Kuthan, U. Abend, H. Schulzrinne, SIP Security, WileyBlackwell, 2009.
[25] P. Park, Voice over IP Security, Cisco Press, 2008.
[26] S. Niccolini, H. Scholz, E. Chen, J. Seedorf, IETF DRAFTv2: SPEERMINT Security Threats and Suggested Countermeasures, Technical Report, IETF, 2010.
[27] J. Rosenberg, C. Jennings, The Session Initiation Protocol (SIP) and Spam, RFC 5039 (Informational), 2008.
[28] S. Bishop, M. Fairbairn, M. Norrish, P. Sewell, M. Smith, K. Wansbrough, Rigorous specification and conformance testing techniques for network protocols, as applied to TCP, UDP, and sockets, SIGCOMM Comput. Commun. Rev. 35 (2005) 265–276.
[29] A. M. Hagalisletto, L. Strand, W. Leister, A.-K. Groven, Analysing protocol implementations, in: F. Bao, H. Li, G. Wang (Eds.), The 5th Information Security Practice and Experience Conference (ISPEC 2009), volume LNCS 5451, Springer Berlin / Heidelberg, 2009, pp. 171–182.