Authentication and Authorisation for Integrated SIP Services in Heterogeneous Environments

Dorgham Sisalem, Jiri Kuthan

Fraunhofer Institute for Open Communication Systems (FhG Fokus)
Kaiserin-Augusta-Allee 31, 10589 Berlin,

Abstract: In order to provide secure and high quality IP-based communication in heterogeneous environments there is a clear need to couple the signalling protocols used for establishing such communication sessions with supporting components and services providing QoS control, security and mediations between different technologies. In this paper we will be investigating the issue of providing an authorization infrastructure for VoIP based sessions that allows the establishment of VoIP sessions and coupling those sessions with a row of supporting services.

1 Introduction

The session initiation protocol (SIP) [1] was primarily designed as a tool for establishing and controlling communication sessions between two or more end systems or users. With this regard, SIP is increasingly being hailed as the standard protocol for VoIP and instant messaging in both the Internet as well as 3G UMTS networks as part of the IP-based multimedia subsystem (IMS).
In a perfect world, having access to an IP network paired with a signalling protocol such as the session initiation protocol (SIP) [1] would be sufficient to establish end-to-end communication between any two users. However, in reality and especially in wireless environments such as UMTS networks, a row of other supporting services is required to transparently establish a communication session between mobile users with an acceptable QoS level. Further, as depicted in Figure 1 various translation and transcoding services are needed to allow the establishment of a communication session in heterogeneous environments. The heterogeneity might be caused by the following factors:

- **End devices:** This includes end devices using different media representation approaches. This involves different compression styles or text or audio capabilities only.
- **Communication protocols:** This involves establishing communication sessions between entities using different protocols for establishing these sessions. This includes establishing a call between a SIP-based device and an ISDN/GSM phone or SIP to H.323.
- **Security policies:** This involves establishing a session between a user in a private IP network and a user in the public Internet for instance.

Sisalem D. and Kuthan J. (2004).
Authentication and Authorisation for Integrated SIP Services in Heterogeneous Environments.
In *Proceedings of the 2nd International Workshop on Security in Information Systems*, pages 88-98
DOI: 10.5220/0002661700880098
Copyright © SciTePress

To overcome this heterogeneity and allow transparent session establishment a row of so-called supporting services is required. These supporting services include the following examples:

- **QoS Establishment Services:**. This indicates mechanisms for providing assured resources in terms of bandwidth for the media sessions established with SIP. Especially in networks with scarce but valuable bandwidth resources such as wireless networks, the session establishment needs to be coupled with the mechanisms that are provided by mobile network operators (MNOs) for ensuring the availability of the needed resources for the session.
- **Connection Services for Heterogeneous Networks:** When contacting users in a non-SIP environment, i.e., users not using SIP as their signalling protocol such as PSTN and GSM users, the SIP signalling needs to be terminated at the one side and translated to the other protocol. Thereby to achieve transparent communication between the users of the two environments a service provider needs to support gateways between these environments.
- **Firewall and Network Address Translation Services:** These services indicate components that are used to protect private networks form attackers as well hide their internal structure. Such components include firewalls and network address translators (NATs). Firewalls usually have a set of fixed rules indicating which ports and addresses can be reached from the outside as well as which addresses and port numbers the users are allowed to connect to from the inside. NATs are used to map a row of private addresses and port numbers to a smaller number of public IP addresses and port numbers. This has effect of hiding the internal addressing structure of the private network and reduces the expenses of buying larger sums of public IP addresses. As SIP users dynamically negotiate addresses and port numbers static firewall rules cannot be used, as the system administrator has no advance knowledge of addresses and port numbers to be used for the communication [2]. Thereby, to allow SIP signalling and media exchange over firewalls and NATs some interaction between the SIP infrastructure and the firewalls and NATs is needed to allow dynamically changing the firewall rules and mirror possible address translations in the SIP messages [3].
- **Media Transcoding and Translation Services:** This type of service can be used to allow users using devices with incompatible compression styles for example to communicate with each other. As a possible supporting service, a service provider might offer translation and transcoding services such as speech to text transcoders to allow a hearing impaired person to contact another person that is using a voice only device.
- **Conference Services:** As a further supporting service, a service provider might offer a conference server for enabling small to medium sized conference sessions. This service might include a media mixer and a centralized conferencing site at which users might login, initiate a session and invite other users to join the conference.

Thereby, providing a SIP-based communication infrastructure implies some sort of integration between the above mentioned services and SIP. This might involve some modification and enhancement of the SIP signalling itself but also a tight correlation

in the authentication, authorization and accounting (AAA) procedures. In this paper we will be investigating the issue of providing authentication and authorization mechanisms for SIP based sessions that allow establishing SIP sessions and coupling those sessions with a row of supporting services. In a first step, see Sec. 2, we will briefly describe the common approaches for authenticating a user's identity. The major part of the work, see Sec. 3, will then be dedicated for describing possible approaches for authorizing a user's request for a service consisting not only of the SIP session but also of supporting services. The described mechanisms will then be evaluated in terms of their applicability, scalability and security among other features in Sec. 4.

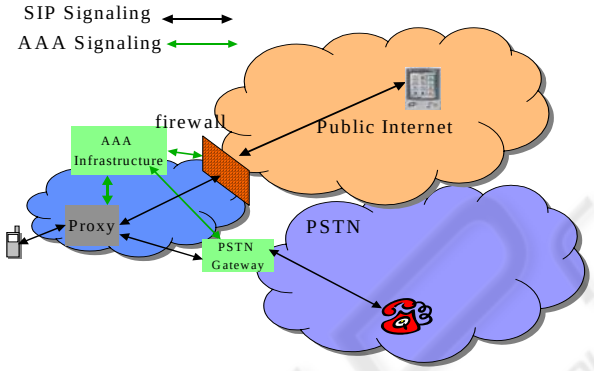

Figure 1. SIP in heterogeneous environments

2 Authenticating Service Requests

The main goal of the authentication procedure is to provide a proof of identity of both the users and providers. For proving the identity of a provider, schemes based on trusted digital certificates are usually used such as with TLS, see [8].

For authenticating users, we can in general distinguish two approaches:

- **Request-based authentication:** With this mechanism the service provider authenticates each request issued by the user. This in general involves a challenge-reply kind of mechanisms such as HTTP Digest, which was specified to be use with SIP.
- **Session-based authentication:** With this approach the authentication procedure is carried out once before the user starts sending any requests. During this phase the user and provider establish a temporary key that can be used to sign and possibly encrypt all requests sent by the user until the termination of the session. UMTS AKA as described in 3GPP [4] present such approaches

For some support services such as QoS for which the user might issue explicit requests as well, similar authentication mechanisms might be used.

3 Coupling SIP Sessions with Supporting Services

When coupling supporting services with a SIP session there are mainly two possibilities for realizing the authentication and authorization actions: SIP dependent and SIP independent authorization. In the SIP dependent scenario, the authorization actions are dealt with as part of the SIP signalling and the information needed for carrying AAA related information are transported as part of the SIP messages. In the SIP independent scenario, the supporting services use their own protocols for carrying out the required AA steps.

3.1 SIP Dependent Authentication and Authorization

In this case the user gets authenticated and authorized to use a supporting service during the session establishment phase using SIP.

3.1.1 User Initiated Services

In this scenario the end user requests explicitly the service. In order to get authorized to use the service the user needs to present some credentials. These credentials are generated during the SIP session establishment and are often called authorization tokens, see [5] and [6] for more details.

Figure 2 shows a simplified message flow in which the user initiates a SIP session and a service such as QoS is coupled with this session.

1. In the first step the user initiates a SIP session by sending an INVITE message indicating that he would like to use QoS resources. This can be indicated through an extension to the session description protocol (SDP), see [9].
2. The proxy might want to check with a AAA server whether the user is eligible for initiating calls with the indicated message content. The AAA server takes its decision based on local policies as well as the user's profile, which governs which services the user is allowed to utilize. In case the user is not eligible for using the service, the invitation is rejected.
3. In case of a positive reply the INVITE message gets forwarded towards the receiver.
4. The reply to the INVITE message indicates the callee's media characteristics and QoS preferences.
5. After receiving the callee's reply, the proxy has the complete information about the IP addresses and port numbers of the communicating end points as well as the media types, compression styles and bandwidth to be used. This information is then used by the AAA server to create an entry for this session. This entry is indexed by an authorization token that identifies the entry as well as the AAA server generating it and is then given to the proxy.
6. The proxy includes the token into the reply and forwards it to the user.
7. The user can issue a service request, e.g. QoS reservation, which includes the authorization token.
8. To authorize the service request, the service control entity, here a QoS router, can use this token to verify the eligibility of the user to request these resources. This is done by contacting the AAA server identified by the token and informing it about the user's wishes and the token delivered by the user. The token is then

used as a reference for the authorization information generated during the SIP session establishment. The answer of the AAA server is then made based on the comparison of the parameters of the requested service and the values contained in the entry generated during the session setup.

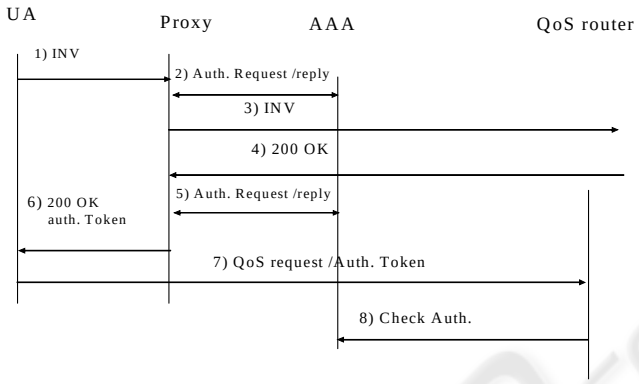

Figure 2. Initiated call in the user initiated services model

In case of session based authentication, the authorization token can be exchanged securely between the SIP proxy either by using an encrypted communication link between the two entities or by encrypting the token using a temporal shared key. The same approach can then be used for exchanging the token between the user and the QoS components. This approach is similar to the one described for 3GPP [4].

In case a mechanisms such as HTTP digest is used for authenticating the user, then the proxy can send the token to the user encrypted with the secret key shared between the user and the SIP provider. The token can then be encrypted with the shared key used between the QoS components and the user for authenticating the user. As the shared keys in this scenario are usually rather short, using tokens in scenarios with request authentication is less secure than for the case of session-based authentication.

3.1.2 Proxy initiated services

In this case the SIP proxy itself initiates the service request and there is no need for exchanging authorization information with the user. This scenario is especially interesting for controlling firewalls and NATs or using a gateway to another network. In this scenario we can distinguish two initiation methods: proxy controlled and proxy routed services.

3.1.2.1 Proxy controlled services

This scenario includes the case for controlling a firewall or a NAT in a midcom like scenario, see [3]. Figure 3 depicts a scenario in which a network is protected by a firewall. This firewall can be controlled by a SIP proxy, which can issue requests to dynamically open certain holes in the firewall and thereby change the filtering rules.

1. After receiving a SIP request the proxy checks with the AAA server whether the user is allowed to make outside calls.
2. If yes, the INVITE gets forwarded to the receiver
3. After receiving the call, the receiver accepts the call and replies with a 200 OK

4. Upon receiving the 200 OK the proxy has the complete information about the addresses and port numbers of the caller and callee. This information is then used to instruct the firewall to change the filtering rules to allow the media traffic of the established session to traverse the firewall.
5. The OK 200 is forwarded
6. Sending an ACK completes the session initiation. Traffic can now flow through the firewall.

Note that in this case no tokens need to be exchanged between the user and the proxy. Thereby both session and request based authentication mechanisms are equal here.

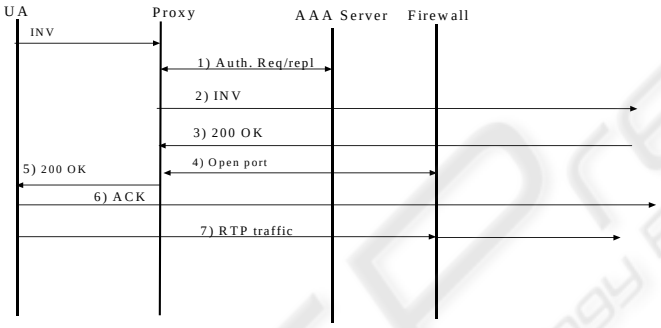

Figure 3 Proxy controlled service

3.1.2.2 Proxy Routed Services

In this scenario, a SIP proxy forwards authenticated and authorized requests to another SIP entity that actually delivers the service. This entity could be a PSTN gateway or some other kind of a transcoding gateway. Figure 4 depicts a scenario in which the user would like to reach a PSTN phone over a gateway.

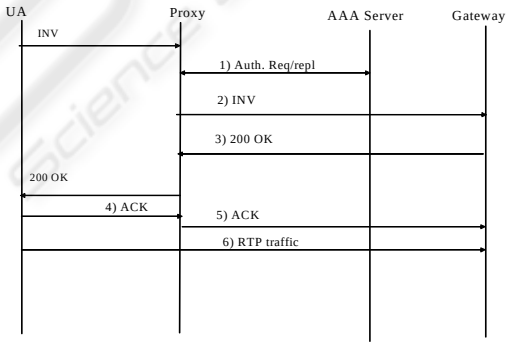

Figure 4 Proxy routed service

1. After receiving a SIP INVITE request for example, the proxy receives the INVITE and checks with the AAA server whether the user is allowed to contact the gateway
2. If the user is authorized to make calls to PSTN destinations the INVITE gets forwarded to the gateway. In this scenario the proxy acts as a kind of a firewall in front of the gateway. Actually it is often the case, that gateways reject all calls not coming from a dedicated proxy. The authenticity of the requests and the assurance that they actually come form a certain trusted proxy, which checks the authorization of the users before forwarding a request, should be guaranteed through a network level security association such as TLS [8] or IPSec between the proxy and the gateway. In order to make sure that all subsequent requests in the session traverse the proxy, the proxy adds a Record-route entry into the INVITE message.
3. The gateway answers with a 200 OK, which is forwarded by the proxy
4. The session establishment is finalized by sending an ACK after which media traffic can be sent to the gateway.

Note that this scenario could also have been realized with the user initiated service scenario, see Sec. 3.1.1. That is the user would receive an authorization token from the proxy and then contact the gateway directly. However, in this case the processing load on the gateway would be increased, as the gateway would need to contact the AAA server to check the correctness of the authorization token.

3.2 SIP Independent Authorization

In this case the user needs to authenticate and authorize himself twice. Once during the SIP session establishment and once during the service request. As depicted in Figure 5 the coupling of the SIP session and the service request is achieved as follows:

1. The user starts the session by issuing a SIP INVITE message.
2. The proxy authenticates and authorizes the user with the help of the AAA server.
3. The INVITE gets forwarded to the destination.
4. The receiver accepts the call by issuing a 200 OK message
5. The OK message gets forwarded to the user.
6. The session establishment is completed by issuing the ACK message.
7. At this stage the user asks for the service.
8. The entity providing the service, e.g., a QoS router, authenticates and authorizes the user. The way this authentication is realized depends very much on the used QoS reservation protocol. For example RSVP proposes the usage of COPs [10] objects, others might use digest authentication similar to SIP.
9. If the user is authorized to use the service then a positive answer is sent.

Notice that the message flow depicted in Figure 5 is only one possibility. As the SIP proxy is not offering expensive services it might not need to authenticate the user at all and thereby we would drop steps 2 and 3. Also, the service request could be established before the SIP session or correlated with it as described in [7].

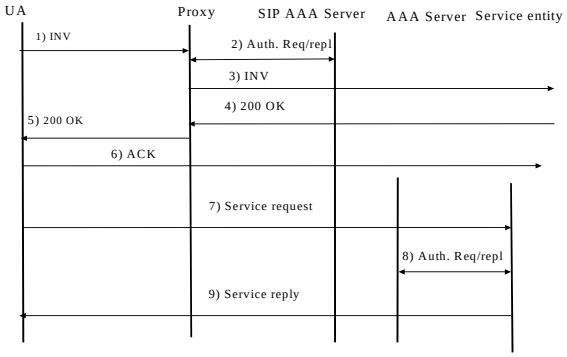

Figure 5. SIP-independent authorization

4 Summary and Conclusions

In this paper we have described various possibilities for enhancing SIP services with a number of supporting services such as QoS, transcoding components and many more. To finalize our work we compare the advantages and disadvantages of the different approaches regarding issues such as performance, security and applicability among others. We will see that choosing the optimal approach for realizing AAA in such a scenario is difficult and depends often on the natures of supporting service.

- **Performance:** In case the user needs to be authorized for both the SIP session and the service usage, the SIP independent approach requires a higher overload in terms of exchanged messages and time. The exact difference depends very much on the authentication mechanisms used by the service entities. For example for the case of the user initiated services and with mechanisms similar to those used for SIP (HTTP DIGEST) we can assume twice the authentication delay and the same time for checking the AAA server. That is in the case of SIP-independent authorization, the service entity would contact the AAA server to check the eligibility of the user. In the case of dependent authorization, the service entity would also need to check the authorization token with the AAA server that generated it. For the case that the SIP session establishment does not require authentication and authorization, both schemes have similar performance. This scenario is especially valid when a user utilizes a public SIP provider which does not require authentication for issuing invitations but still wants to use the QoS infrastructure provided by the network access provider.
- **Applicability:** The applicability of both SIP dependent and independent authorization to the different service scenarios identified in Sec. 2 depends greatly on the service.
  - **QoS establishment service:** Both approaches are applicable to the scenario of coupling QoS reservations with a SIP session. For the case of SIP independent authorization, the QoS protocols need, however, to incorporate

user authentication and authorization more closely with the QoS reservation protocols. This would further increase the complexity of such protocols. With the dependent approach, either the proxies can instruct the QoS components to provide certain QoS features, or the QoS protocols would carry the authorization tokens.

- **Network security and translation service:** For the case of traversing a firewall, SIP independent authorization does not apply easily as controlling the middle box requires knowledge about all the communicating end systems. NAT traversal is not possible with the SIP independent scenario, as the SIP proxy needs to know the results of the address translation of the media flows already during the signalling phase.
- **Connection services:** For the case of gateway usage, using the SIP dependent scenario is preferred. The SIP proxy provides a kind of a firewall in front of the gateway filtering unauthorized requests and reducing the load on the gateways that would have been otherwise required to authenticate and authorize the users. The SIP independent scenario is applicable as well but would require the user to authenticate himself directly with the gateway. This would imply, that the gateway needs to maintain its own AAA infrastructure and relation with the user.

- **Security:** This aspect indicates whether the used solution would have negative effects on the security of the communication session or the signalling protocol. Also we need to avoid introducing new possibilities for denial of service attacks or data manipulation
  - For a proxy to authorize a QoS reservation for example, it needs to extract the media description data from the SIP messages and analyze them. This means that SIP messages cannot use end-to-end encryption in the SIP dependent scenario. This is not an issue for the SIP independent scenario.
  - Another aspect is the security of the exchanged authorization token between the proxy and the user in the user initiated scenario. This data usually indicates the entity that generated the token as well as a special entry to the authorization data generated at that entity during the SIP signalling. Stealing this data could allow an interceptor to generate QoS requests under the identity of the actual user involved in the SIP session establishment. As described in Sec. 3.1.1, this can be avoided by encrypting the exchanged tokens. Further, this risk can be reduced by indicating in the AAA entries created during the session establishment phase the exact addresses of the communicating entries. Thereby during the QoS reservation phase only reservations between those addresses can be established. However, this still allows for a denial of service attack. By sending data to the callee and putting the IP address of the caller in the data packets an attacker can reduce the share used by the actual caller of the QoS resources and thereby incur costs on him for resources he did not use. To avoid this case, the communication link between the proxy sending the authorization token and the end systems needs to be secured. This involves establishing a shared key between the involved entities and signing sent packets with this key, which further complicates the session set-up and initial authentication procedures. Another

option would be protect the token so that only the end system that has initiated the session can decipher it. In this case, the SIP provider would encrypt the token using a key shared with the user. The user would decrypt the token and add it to his QoS reservation request. For a better protection, the user might again encrypt the token using a key shared with the QoS provider.

- **Complexity:** The aspect of complexity describes here the changes needed to existing components and the additional overhead required for managing new components.
  - For a proxy to authorize a QoS reservation for example, it needs to not only parse and understand the headers of the SIP message but also the session description part as well. This increases the complexity of the SIP proxy and increases the processing overhead.
  - Another aspect is the authorization part itself. In case the user is authenticated and authorized using SIP then the protocols needed for requesting the supporting services might be simplified and do not require such mechanisms.
  - The token mechanism requires extensions to both SIP as well as the service protocols with headers to include the token. Further, the end user needs to coordinate the usage of both SIP and the supporting service by taking the token from SIP and adding it to the service signalling part.
- **Flexibility:** For supporting SIP-dependant service coupling, there always needs to be some trust relation between the SIP provider and the service provider. This can take the form of a secure connection or might be realized using a trusted AAA infrastructure. Thereby, in order to provide new services, the new service provider needs first to establish this trust relation with the SIP provider. This might lead to delays in the introduction of the service or creating dependencies that might make the entry of new service providers more difficult. With the SIP independent AAA scenario, there is no need for a trust relation between the SIP provider and the service provider. However, this scenario requires trust relations between the user and the service provider.
- **Convenience:** The SIP dependent authorization scenario has the big advantage that the user only needs to establish a trust relation with one provider and is only presented with one bill for the resources he is using. With the SIP independent authorization scenario, the user would need to maintain a contractual relation to the providers of each supporting service he would like to use and establish a new relation for each new service.

References

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Spark, M. Handley, E. Schooler, “Session Initiation Protocol”, RFC3261, June 2002

[2] M. Holdrege and P. Srisuresh "Protocol complications with the IP network address translator (NAT)", RFC 3027, January 2001.

[3] P. Srisuresh, J. Kuthan, J. Rosenberg: “Middlebox Communication Architecture and framework”, February 2001, IETF, Internet Draft
[4] 3GPP Technical Specification 3GPP TS 33.102 V3.6.0: "Technical Specification Group Services and System Aspects; 3G Security; Security Architecture (Release 1999)", 3rd Generation Partnership Project, November 2000
[5] W. Marshall, F. Andreasen, D. Evans, “SIP Extensions for Media Authorization”, Internet draft, May, 2002
[6] Sinnreich, Rawlins, Gross, Thomas, "QoS and AAA Usage with SIP based IP communication", Internet Draft, Internet Engineering Task Force, October 2001
[7]Camarillo, Marshall, Rosenberg, “Integration of Resource Management and SIP”, Internet Draft, April 2002
[8] Dierks, C. Allen, "The TLS Protocol Version 1.0" RFC 2246, January 1999.
[9] M. Handley and V. Jacobson, "SDP: session description protocol," RFC 2327, Internet Engineering Task Force, Apr. 1998
[10] Boyle, J., Cohen, R., Durham, D., Herzog, S., Raja, R. and A. Sastry. "The COPS (Common Open Policy Service) Protocol", RFC 2748, January 2000