Internet Engineering Task Force MMUSIC WG
INTERNET-DRAFT
draft-ietf-mmusic-sap-00.ps
Mark Handley
ISI
19th Nov 1996

SAP: Session Announcement Protocol

Status of this Memo

This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as ‘‘work in progress.’’

To learn the current status of any Internet-Draft, please check the ‘‘1id-abstracts.txt’’ listing contained in the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

Distribution of this document is unlimited.

Abstract

This document describes the SAP - the session directory announcement protocol, and the related issues affecting security and scalability that should be taken into account by the implementors of session directory tools. It is a companion document to draft-ietf-mmusic-sdp.

This document is a product of the Multiparty Multimedia Session Control (MMUSIC) working group of the Internet Engineering Task Force. Comments are solicited and should be addressed to the working group’s mailing list at confctrl@isi.edu and/or the author.

1. Introduction

An mbone session directory is used to advertise multimedia conferences, and to communicate the session addresses (whether multicast or unicast) and conference-tool-specific information necessary for participation. Such sessions are described using the Session Description Protocol (SDP) which is described in a companion draft. This document describes the issues involved in the multicast announcement of session description packets and defines a packet format to be used by session directory clients. SAP v0 is currently implemented in Sdr and other compatible tools. This document describes SAP v1, which contains some enhancements to the basic announcement model. The differences between SAP v1 and SAP v0 are described in Appendix A. *Much of this document is concerned with security considerations - these security considerations have not yet been subject to suitable peer-review, and this document should not be considered authoritative in this area.*

2. Background

IP Multicast is an extension of internet routing that permits efficient many-to-many communication. It is used extensively for multimedia conferencing. Such multimedia sessions usually have the property that tight coordination of session membership is not necessary; in order to receive a session, a user at a multicast-capable site only has to know the correct multicast group address for the session and the transport ports the conferencing applications will use to receive the conference data streams.

In order to assist the advertisement of multicast sessions and to communicate the relevant session setup information to prospective participants, a distributed *session directory* is used. An instance of such a session directory periodically multicasts packets containing a *description* of a multimedia session, and these advertisements are received by potential participants who can use the session description to start the tools required to participate in the session. The companion draft ‘‘SDP: Session Description Protocol’’ describes a payload format suitable for such session descriptions. This draft describes the distribution mechanism and packet format.

3. The SAP Protocol

SAP is an announcement protocol for multicast conference sessions. An SAP client that announces a conference session periodically multicasts an announcement packet to a well known multicast address and port. The announcement is multicast with the same scope (as defined by group address range or TTL) as the session it is announcing. This ensures that the recipients of the announcement can also be potential recipients of the session the announcement describes (bandwidth and other such constraints permitting). This is also important for the scalability of the protocol, as it keeps local session announcements local.

The time period between one announcement and its repetition is dependent on two factors - the scope (TTL) of the session, and the number of other sessions currently being announced by other session directory clients. The goal is to keep the total bandwidth being used below a predefined level for each scope.

Session Announcement

A session to be announced is simply multicast to the appropriate well-known multicast address and port. The announcement contains a session description and, optionally, an authentication header. The session description may be encrypted.

Session Deletion

Sessions may be deleted in one of several ways:

Explicit Timeout

> The session description contains timestamp information which specifies a start and end time for the session. If the current time is later than the end-time for the session, then the session is deleted from the receiver’s session cache. If an announcement packet arrives with an end-time before the current time, it is ignored.

Implicit Timeout

> A session announcement message should be received periodically for each session description in a receiver’s session cache. The announcement period can be predicted by the receiver from the set of sessions currently being announced. If a session announcement message has not been received for ten times the announcement period, or half an hour, whichever is the

greater, then the session is deleted from the receiver's session cache. The half hour minimum is to allow for transient network partitionings.

Explicit Deletion

A session deletion packet is received specifying the *version* of the session to be deleted. If the cached session contains an authentication header, the session deletion packet must contain a signature signed by the same key. If the cached session does not contain an authentication header, but the deletion packet has the same IP-source address (*not* the SAP-stated source address in the packet) as that from which the session announcement was originally announced, then the session is deleted. If neither of these conditions is not the case, then the session deletion packet is ignored. Note that IP source addresses can be spoofed, and although the RPF filter in most multicast routing algorithms will result in the packet not being delivered in some cases, this is insufficient protection in many cases, and an authentication header should be used for such situations.

Session Modification

A pre-announced session can be modified by simply announcing the modified session description. In this case, the version hash in the SAP header should be changed to indicate to receivers that the packet contents should be parsed (or decrypted and parsed if it is encrypted). The session itself is uniquely identified by the SDP origin field in the payload, and not by the version hash in the SAP header.

The same rules apply for session modification as for session deletion:

- Either the modified announcement must contain an authentication header signed by the same key as the cached session announcement it is modifying, or:
- The cached session announcement *must not* contain an authentication header, and the session modification announcement must originate from the same host as the session it is modifying.

If an announcement is received containing a authentication header and the cached announcement did not contain an authentication header, or it contained an different authentication header, then the modified announcement must be treated as a new and different announcement, and displayed *in addition* to the un-authenticated announcement. The same should happen if a modified packet without an authentication header is received from a different source than the original announcement. These rules prevent an announcement having an authentication header added by a malicious user and then being deleted using that header, and it also prevents a denial-of-service attack by someone putting out a spoof announcement which, due to packet loss, reaches some participants before the original announcement. Note that under such circumstances, being able to authenticate the message originator is the only way to discover which session is the correct session.

4. Packet Format

Unencrypted SAP data packets are of the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| V=1 | MT  |E|C|   auth len    |         msg id hash           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          orig source                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 optional authentication header                |
|                              ....                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          text payload                         |
|                              ....                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Encrypted SAP data packets contain additional fields

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| V=1 | MT  |E|C|   auth len    |         msg id hash           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          orig source                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 optional authentication header                |
|                              ....                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              key id                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             timeout                           |
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
|P|                         random field                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          text payload                         |
|                              ....                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Only fields from `*` onwards are encrypted.

V: Version Number

SAP version number = 1

MT: Message Type

One of the following:

0 **Session description announcement packet.** The text payload is an SDP session description, as described in draft-ietf-mmusic-sdp.

1 **Session description deletion packet.** The text payload is a single SDP line consisting of the origin field of the announcement to be deleted.

E - Encryption Bit

If the encryption bit is set, the text payload of the SAP packet is encrypted, and additional fields are added to the packet: Key-ID, Timeout, P (padding) and Random. The Key-ID and Timeout fields are not encrypted, but the P and Random fields are encrypted along with the text payload. Note the encryption algorithm is not specified in the packet - this is communicated to permitted receivers out-of-band along with the corresponding decryption key.

C - Compressed bit

This bit indicates that the payload was compressed using the gzip compression algorithm [3].

Authentication Length:

A 8 bit unsigned quantity giving the number of 32 bit words following the main SAP header that contain authentication data (and padding bytes if present). If it is zero, no authentication header is present.

Authentication Header

This contains a digital signature (encrypted cryptographic hash) of the text payload (including key-id, expiry timestamp, and encrypted random field and text payload if the payload is encrypted) from the end of the authentication header onwards. It also contains the public key with which the authentication header can be checked, and information to identify the encryption algorithm and mode used. It can be used for two purposes:

- Verification that changes to a session description or deletion of a session are permitted.
- Authentication of the identity of the session creator.

In some circumstances only Verification is possible because a certificate signed by a mutually trusted person or authority is not available. However, under such circumstances, the session originator may still be authenticated to be the same as the session originator of previous sessions claiming to be from the same person. This may or may not be sufficient depending on the purpose of the session and the people involved.

Clearly the key given in the authentication header should not be trusted to belong to the session originator unless it has been separately authenticated by some other means, such as being certified by a trusted third party. Such certificates are not normally included in an SAP header because they take more space than can normally be afforded in an SAP packet, and such verification must therefore take place by some other mechanism. However, as certified public keys are normally locally cached, authentication of a particular key only has to take place once, rather than every time the session directory retransmits the announcement.

SAP is not tied to any single authentication mechanism. Authentication Headers must be self-describing, but their precise format depends on the authentication mechanism (signature and encryption scheme) in use, and so is not defined here.

Message Identifier Hash

A 16 bit quantity that, used in combination with the originating source, provides a globally unique id identifying the precise version of this announcement. The message id hash should be changed if any field of the session description is changed. A value of zero means the hash should be ignored and the message should always be parsed.

Originating Source

This gives the IP address of the original source of the message. It is permissible to be zero if the message has not passed through a proxy relay and if the message id hash is also zero, though this is intended only for backwards compatibility with SAPv0 clients.

Key ID

The key identifier is a 32 bit network byte-order integer which is used as a hint to identify which encryption key was used to encrypt a packet. Key id's should be randomly generated when a new encryption key is chosen for a group of users, and so they are not guaranteed to be globally unique. If a receiver has multiple keys with the same key-id, to perform decryption each key in turn must be used until one of them successfully decrypts the data.

Timeout

When the session payload is encrypted, and the session description is being relayed or announced via a proxy, the detailed timing fields in the SDP description are not available to the proxy as they are encrypted and the proxy is not trusted with the decryption key. Under such circumstances, SAP includes an additional 32-bit timestamp field stating when the session should be timed out. This field is included after the authentication header, and the digital signature in the authentication header encompasses the timeout so that a session cannot be maliciously deleted by modifying its timeout in an announcing proxy.

The value is an unsigned quantity giving the NTP time [2] in seconds at which time the session is timed out. It is in network byte order.

P: Encryption Padding

This bit indicates that the payload was padded prior to encryption. The last byte of the decrypted payload indicates how many padding bytes were added.

Random

This field is only present when the payload is encrypted. It is encrypted along with the payload, and is used to perform the randomization task normally performed by an initialization vector in algorithms such as cipher-block chained DES. This 31 bit field should contain a genuinely random number. After decryption, this field is discarded.

5. Encrypted Announcements

Announcements may be encrypted using any encryption algorithm or mode. However, the use of DES in cipher-block chaining (CBC) mode is recommended as the default case. The choice of encryption algorithm and mode is conveyed to potential recipients along with the encryption key itself and a 32 bit key identifier which should be randomly chosen and is used as a non-globally-unique identifier for the key.

In normal usage, a {decryption-key,keyid,algorithm,mode} tuple will be conveyed in advance to the intended group recipients. This process takes place out-of-band and is not described in this draft. However, if keys are to be communicated as plain text, the use of MD5 as described in [4] is recommended to manipulate the key prior to use.

Session announcements may then be made to the appropriate session announcement address, encrypted so that they can be decrypted with the group key. The key-id is carried in the announcement packets, and serves as an index into a sparse key-ring at each receiver. As key-id's are allocated randomly, in some cases more than one decryption-key may be identified by the same key id - this is expected to be a rare event, but may happen. When more that one key is identified by a key-id, each of the decryption keys in turn must be tried.

5.1. Encrypting Announcements

If the payload is to be compressed, this is performed first before encryption or padding.

When an announcement is to be encrypted, a 32-bit word is prepended to the session description payload. The most significant bit of this number (in network byte order) is set to zero if the session description does not require padding for encryption, and set to one if padding is required for encryption, in which case the last byte of the padded session description contains the number of padding bytes added. The least significant 31 bits of this 32 bit quantity should contain random data.

The padded and 32-bit pre-pended session description is then encrypted using the relevant encryption algorithm, key and mode. The encrypted payload is then sent with an extended SAP header which has the E bit set and contains the key id and timeout fields as described above.

5.2. Decrypting Announcements

Upon receiving a new announcement with the encryption bit set, a receiver should attempt to decrypt the announcement with each of its set of session decryption keys that has the key-id appearing in the announcement. If it succeeds, then the session is displayed to the user. If it has no key that matches the announcement key-id, or does not succeed with any key that does match, then the session is ignored. If one of more keys did match the key-id, but decryption failed with all of these matching keys, then the version hash, original source and key-id are cached to avoid having to attempt to decrypt this announcement every time it is received in future. To avoid possible denial of service attacks, such incoming announcements should occasionally be attempted to be decrypted on a random basis as available processing power allows. This cache can be safely timed out when the timeout specified in encrypted packets expires.

Note that if an encrypted announcement is being announced via a proxy, then there may be no way for the proxy to discover that the announcement has been superseded, and so it may continue to relay the old announcement in addition to the new announcement. SAP provides no mechanism to chain modified encrypted announcements, so it is advisable to announce the unmodified session as deleted for a short time after the modification has occurred. This does not guarantee that all proxies have deleted the session, and so receivers of encrypted sessions should be prepared to discard old versions of session announcements that they may receive (as identified by the SDP version field). In most cases however, the only stateful proxy will be local to (and known to) the sender, and an additional (local-area) protocol involving a handshake for such session modifications can be used to avoid this problem.

6. SDP announcement by periodic multicast.

SAP announces multicast sessions by periodic multicast of session descriptions to an appropriate well known multicast address and port. The appropriate address is determined by the scope mechanisms in force at the sites of the intended participants. IP multicast sessions can be either TTL-scoped or administratively scoped. One well-known address and port is used for all TTL-scoped announcements, and additionally, one well-known address (within the corresponding scope zone) and port is used for each administrative scope zone that an instance of the session directory is within. Thus an instance of the session directory should *listen* on multiple multicast addresses, but should normally only *send* a particular announcement to the single multicast address corresponding to the scope of the session being described. The discovery of administrative scope zones and the appropriate announcement address for each zone are outside the scope of this draft, but it is assumed that each instance of the session directory within a particular scope zone is aware of that scope zone, and of the corresponding announcement address, port, TTL, and session address allocation range.

TTL Scoped Announcement

The well-known address is 224.2.127.254 and the UDP port is 9875. The session announcements should be multicast with the same TTL with which the conference session will be multicast. If the different media in an announcement are given different TTLs, then multiple announcements are necessary to ensure that anyone joining the conference can in fact receive data for each media started. For example, if we have an announcement to make containing audio at TTL 127 and video at TTL 63, then we make an announcement at TTL 63 containing both media, and a separate announcement at TTL 127 containing only the audio. It is up to the receiving session directory to parse both announcements as the same announcement (as identified by the SDP origin field) if it is within the appropriate scope to get both announcements. If multiple announcements are being made for the same session in this way, then each announcement must carry an authentication header signed by the same key, or be treated as a completely separate announcement.

The time period between one announcement and its repetition is dependent on two factors - the scope (TTL) of the session, and the number of other sessions currently being announced by other session directory instances.

The recommended bandwidth limits for each TTL are:

```
TTL     bandwidth
1-15     2Kbps
16-63    1Kbps
64-127   1Kbps
128-255 200bps
```

Session announcers in the same scope band can normally be expected to hear your announcements, and reduce their data rates accordingly. Thus you should calculate the available bandwidth for your session's scope band by dividing the appropriate limit above by the number of other announcers in your scope band. This gives you your bandwidth allocation, which, given the size of your data packets, can be used to derive the base interval for announcements.

I.e., given a *limit* in bits/second (as above) and a *ad_size* in bytes, the base announcement *interval* in seconds is:

$$interval = \mathrm{MAX}(300, (8*no\_of\_ads*ad\_size)/limit)$$

For every interval between announcement packets (i.e, every time you send a packet), you must add a random value (+/- 1/3 of the base interval) to the value used for the inter-announcement period to prevent announcement synchronisation. It is also important to keep monitoring other announcements and adjust the base interval accordingly.

There is possibility to adjust the scope band limits depending on properties of the sessions being announced, but this is left for future SAP drafts to specify.

Administrative Scoped Announcements

For each administrative scope zone in force at a particular site, instances of the session directory running at that site need to know the following information:

- The multicast address to be used for announcement. The is normally the highest multicast address in the relevant administrative scope zone. For example, if the scope range is 239.16.32.0 - 239.16.33.255, then the convention is that 239.16.33.255 is used for session announcements.
- The UDP port to which announcements should be sent.
- The TTL announcements should be made with. This should be large enough to reach all sites in the admin scope zone, and will also be the TTL used for sessions announced to be using this scope zone.
- The address range to be used for sessions in this scope zone. This should be a contiguous range, and currently should lie within the range 239.0.0.0 to 239.255.255.255 (but this is defined by IANA, not by this draft).
- The total bandwidth to be used by the session directory for session announcements in this admin scope zone. A recommended default value for this is 500bps, but this may be inappropriate for some uses.

7. Security Considerations

SAP contains mechanisms for ensuring integrity of session announcements, for authenticating the origin of an announcement and for encrypting such announcements. *These mechanisms have not yet been subject to suitable peer-review, and this document should not be considered authoritative in this area at this time.*

SAP contains mechanisms that are designed to prevent an announcement by one user from being modified or deleted by another user, and also to provide limited privacy by use of encryption.

Session announcements that are encrypted with a symmetric algorithm may allow a degree of privacy in the announcement of a session, but it should be recognised that a user in possession of such a key can pass it on to other users who should not be in possession of such a key. Thus announcements to such a group of key holders cannot be assumed to have come from an authorised key holder unless there is an appropriate authentication header signed by an authorised key holder. In addition the recipients of such encrypted announcements cannot be assumed to only be authorised key holders. Such encrypted announcements do not provide any real security unless all of the authorised key holders are trusted to maintain security of such session directory keys. This property is shared by the multicast session tools themselves, where it is possible for an untrustworthy member of the session to pass on encryption keys to un-authorised users. However it is likely that keys used for the session tools will be more short lived that those used for session directories.

Similar considerations should apply when session announcements are encrypted with an assymetric algorithm, but then it is possible to restrict the possessor(s) of the private key, so that announcements to a key-holder group can not be made, even if one of the untrusted members of the group proves to be un-trustworthy.

As stated above, if a session modification announcement is received that contains a valid authentication header, but which is not signed by the original creator of the session, then the session must be treated as a new session *in addition* to the original session with the same SDP origin information unless the originator of one of the session descriptions can be authenticated using a certificate signed by a trusted third party. If this were not done, there would be a possible denial of service attack whereby a party listens for new announcements, strips off the original authentication header, modifies the session description, adds a new authentication header and re-announces the session. If a rule was imposed that such spoof announcements were ignored, then if packet loss or late starting of a session directory instance caused the original announcement to fail to arrive at a site, but the spoof announcement did so, this would then prevent the original announcement from being accepted at that site.

A similar denial-of-service attack is possible if a session announcement receiver relies completely on the originating source and hash fields to indicate change, and fails to parse the remainder of announcements for which it has seen the origin/hash combination before.

A denial of service attack is possible from a malicious site close to a legitimate site which is making a session announcement. This can happen if the malicious site floods the legitimate site with huge numbers of (illegal) low TTL announcements describing high TTL sessions. This may reduce the session announcement rate of the legitimate announcement to below a tenth of the rate expected at remote sites and therefore cause the session to time out. Such an attack is likely to be easily detectable, and we do not provide any mechanism here to prevent it.

Appendix A: Summary of differences between SAPv0 and SAPv1

For this purpose SAPv0 is defined as the protocol in use by version 2.2 of the Sdr session description tool. SAPv1 is the proposed protocol described in the document. The packet headers of SAP messages are the same in V0 and V1 in that a V1 tool can parse a V0 announcement header but not vice-versa.

In SAPv0, the fields have the following values:

- Version Number: 0
- Message Type: 0 (Announcement)
- Authentication Type: 0 (No Authentication)
- Encryption Bit: 0 (No Encryption)
- Compression Bit: 0 (No compression)
- Message Id Hash: 0 (No Hash Specified)
- Originating Source: 0 (No source specified, announcement has not been relayed)

Appendix B: Author's Address

Mark Handley
Information Sciences Institute,
University of Southern California,
c/o MIT Laboratory for Computer Science,
545 Technology Square,
Cambridge, MA 02139,
United States
electronic mail: mjh@isi.edu

Acknowledgments

SAP and SDP were originally based on the protocol used by the sd session directory from Van Jacobson at LBNL. The design of SAP was funded by the European Commission under the Esprit 7602 "MICE" project, and the Telematics 1007 "MERCI" project.

References

[1] M.Handley, V. Jacobson, ‘‘SDP: Session Description Protocol’’, INTERNET-DRAFT, Nov 1996.

[2] D. Mills, ‘‘Network Time Protocol version 2 specification and implementation", RFC1119, 1st Sept 1989.

[3] P. Deutsch, ‘‘GZIP file format specification version 4.3’’, RFC 1952, May 1996.

[4] H. Schulzrinne, ‘‘RTP Profile for Audio and Video Conferences with Minimal Control’’, RFC 1890, January 1996